Till file

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Internal Note No.
69-FM-232

MSC INTERNAL NOTE NO. 69-FM-232

September 2, 1969

DESCRIPTION OF APOLLO GUIDANCE ANALYSIS STATISTICAL TRIALS PROGRAM

Orbital Mission Analysis Branch

MISSION PLANNING AND ANALYSIS DIVISION

MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

(NASA-TM-X-69638) DESCRIPTION OF APOLLO
GUIDANCE ANALYSIS STATISTICAL TRAILS
PROGRAM (NASA) 531 p

N74-70856

Unclas
00/99 16226

MSC INTERNAL NOTE NO. 69-FM-232

DESCRIPTION OF APOLLO GUIDANCE ANALYSIS STATISTICAL TRIALS PROGRAM

By Elric N. McHenry, Dennis M. Braley,
Ernest M. Fridge, and Robert N. Hinson
Orbital Mission Analysis Branch

September 2, 1969

MISSION PLANNING AND ANALYSIS DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
HOUSTON, TEXAS

Approved: Edgar C. Lineberry
Edgar C. Lineberry, Chief
Orbital Mission Analysis Branch

Approved:
John P. Mayer, Chief
Mission Planning and Analysis Division

CONTENTS

Section Page

1.0 SUMMARY 1

2.0 INTRODUCTION 1

3.0 DEFINITIONS 3

3.1 Abbreviations 3

3.2 Algebraic Symbol Definition 4

4.0 COORDINATE SYSTEMS 8

4.1 Apollo Local Vertical Frame 8

4.2 The UVW Coordinate System 8

4.3 Line-of-sight Coordinate System 8

4.4 Nominal Platform Alinement Coordinate System 9

4.5 Preferred Platform Alinement Coordinate System 9

4.6 Landing Site Platform Alinement Coordinate System 9

5.0 TRAJECTORY ERROR ANALYSIS TECHNIQUE 10

5.1 Statistical Trials Technique 10

5.2 AGAST Trajectory Simulation 11

6.0 PROGRAM CAPABILITIES 15

6.1 Computer Monitoring 15

6.2 Error Sources Modeled 15

6.3 Integrators and Central Body Options 16

6.4 Program Initialization 16

Section Page

APPENDIX B - FLOW CHARTS OF THE SUBROUTINE OF AGAST 93

APPENDIX C - DEFINITIONS OF SYMBOLS USED IN SUBROUTINES OF AGAST 461

REFERENCES 524

DESCRIPTION OF APOLLO GUIDANCE ANALYSIS

STATISTICAL TRIALS PROGRAM

By Elric N. McHenry, Dennis M. Braley,
Ernest M. Fridge, and Robert N. Hinson

1.0 SUMMARY

The purpose of this report is to describe the techniques, capabilities, and formulations of the Apollo Guidance Analysis Statistical Trials Program. The Apollo Guidance Analysis Statistical Trials Program (AGAST) is an error analysis computer program designed to aid studies of Apollo rendezvous mission plans. The AGAST simulates targeting, guidance, and navigation computations specified for the Apollo rendezvous missions. Error analyses may be performed by AGAST for both lunar and earth missions. The error analysis technique used is that of statistical trials, that is, the Monte Carlo method (ref. 1).

2.0 INTRODUCTION

A collection of FORTRAN language simulations of routines in the lunar module (LM) and command/service module (CSM) onboard computers form the core of the program. Modeled are the onboard guidance and navigation systems of the command module computer (CMC), the LM primary guidance and navigation control system (PGNCS), and the LM abort guidance system (AGS) as documented in references 2, 3, and 4, respectively. The maneuver targeting and guidance equations for LM descent to the lunar surface, LM ascent into lunar orbit, and the concentric sequence rendezvous are programed into AGAST. Other onboard routines coded into AGAST are the state vector integrators and the navigation equations. The program has the capability to simulate both AGS and PGNCS computations and maneuver control. Also modeled are some of the Real-Time Computer Complex (RTCC) rendezvous maneuver targeting computations and Manned Space Flight Network (MSFN) state vector updates. To perform an error analysis of a mission profile, the program may model any or all of the four computers: CMC, PGNCS, AGS, and RTCC.

The program simulates a mission by performance of maneuver computations (targeting and guidance) and integration of the state vectors between maneuvers. The update of the state vector estimates by either onboard sensors or ground based tracking (MSFN) is also simulated. The mission plan profile is defined by input packages to the program. Each individual input package is called an event package and defines a specific set of computations to be performed. Program computations are actuated by the reading of the event packages on computer cards (ref. 5). The program user may have the program simulate events of the following kinds.

a. Maneuver events

b. Vehicle-to-vehicle transferral of maneuver information

c. Platform alinement

d. AGS-to-PGNCS alinement

e. Vehicle-to-vehicle update of state vector estimates

f. MSFN update of state vector estimates

g. Update of onboard state vector estimates with onboard sensors (rendezvous radar, etc.)

There are few program restrictions on the order or number of events which define the mission profile; the program is very flexible and requires little or no modification for the different Apollo missions.

Although AGAST has been developed to be used primarily to perform the integrating Monte Carlo trajectory error analysis, it need not necessarily be used for that purpose; it may be used to perform deterministic (error free) studies of a variety of mission plans. It may also be used to perform studies of the effects of particular error sources; for instance, the program user may specify certain engine performance discrepancies and may obtain the deterministic effects of those discrepancies on the mission.

Documentation may be found in the references mentioned in appendix A for all subroutines which are used in AGAST but not represented by a flow chart in appendix B. Symbols used in appendices A and B are defined in appendix C. The authors wish to express their appreciation for the help rendered by the ITT-FEC engineering aides in the preparation of the figures and flow charts in this document.

3.0 DEFINITIONS

3.1 Abbreviations

AGAST	Apollo Guidance Analysis Statistical Trials program
AGS	abort guidance system
APS	ascent propulsion system
BR	basic reference coordinate system
CMC	command module computer
CSM	command/service module
DPS	descent propulsion system
IMU	inertial measurement unit
LGC	lunar module guidance computer
LM	lunar module
LOS	line-of-sight coordinate system
LS	landing site
LV	local vertical coordinate system
MSFN	Manned Space Flight Network
NA	nominal alinement
PA	preferred alinement
PGNCS	primary guidance and navigation control system
RCS	reaction control system
REFSMMAT	rotation matrix which defines the IMU orientation in basic reference coordinates
RR	rendezvous radar
RTCC	Real-Time Computer Complex

SCT	scanning telescope
SXT	sextant
VHF	relative range measuring device on the CSM

3.2 Algebraic Symbol Definition

$[A]$	rotation matrix maintained by the AGS which relates the AGS strapped down platform to the AGS reference frame
$[B]$	rotation matrix that represents the bias misalinement of the accelerometers with respect to the platform
$[I]$	the n by n identity matrix
R_A	actual relative range between two vehicles
R_M	measured relative range between two vehicles
$\dot{R}_A$	actual relative range rate between two vehicles
$\dot{R}_M$	measured relative range rate between two vehicles
$[P(t)]$	IMU platform orientation error matrix; rotates from ideal IMU orientation to the true orientation of the IMU
$[P_T]$	value of $[P(t)]$ at the time of an AGS-to-PGNCS alinement
$[S]$	diagonal matrix with the accelerometer scale factors of each accelerometer axis on the diagonal
$[T]$	REFSMMAT; rotation matrix from basic reference coordinates to the ideal orientation of the platform
t	current time
$\bar{u}_T$	unit vector along the thrust direction
$\bar{u}_{TA}$	unit vector of the actual thrust direction
$\bar{u}_{TD}$	unit vector of the desired thrust direction

$\bar{V}_G$	velocity-to-be-gained for burn
X	six-dimensional state vector of a vehicle's position and velocity in BR coordinates
X_A	actual (environmental) state vector
X_E	estimated (onboard or MSFN) state vector
X_E^*	value of the onboard state being updated which results in no estimated relative state errors in a local sense
X_R	reference (nominal) state vector
$\bar{x}$	n-dimensional vector subject to random Gaussian dispersions
$\left\{\bar{x}_i\right\}_{i=1}^{N}$	array of N random samples of the random vector $\bar{x}$
β_A	actual RR or SXT shaft angle
β_M	measured RR or SXT shaft angle
$\Delta\bar{V}_A$	actual incremental velocity change added during a guidance cycle
$\Delta\bar{V}_B$	accelerometer $\Delta\bar{V}$ read bias
$\Delta\bar{V}_{co}$	burn error due to pilot cutoff error
$\Delta\bar{V}_E$	measured $\Delta\bar{V}$ rotated into BR coordinates
$\Delta\bar{V}_M$	accelerometer measured $\Delta\bar{V}$ in platform coordinates
$\Delta\bar{V}_P$	burn error due to initial thrust vector misalinement
ΔX	six-dimensional relative state vector between two vehicles; for instance, the nominal relative state vector is: $\Delta X_R = X_{R,I} - X_{R,J}$ where the relative position vector is directed from vehicle J to vehicle I

ΔX_A	actual relative state vector
ΔX_E	estimated relative state vector
ΔX_E^*	estimated relative state which the onboard tracking must produce to have no errors in a local vertical sense
δR	measurement bias in R_M
$\delta \dot{R}$	measurement bias in $\dot{R}_M$
δX_A	difference between X_A and X_R
δX_E	difference between X_E and X_A
δX_{REL}	error in the estimated relative state vector
$\delta \bar{x}$	n-dimensional random dispersion of $\bar{x}$ whose components have zero mean; i.e., $\bar{x} = \bar{\mu} + \bar{\delta} x$
$\delta\beta$	measurement bias in β_M
$\delta\theta$	measurement bias in θ_M
$\bar{\eta}$	n-dimensional random vector whose components individually have zero mean and unity variance; i.e., a normalized random vector
θ_A	actual RR or SXT trunnion angle
θ_M	measured RR or SXT trunnion angle
$\bar{\mu}$	mean of the random vector $\bar{x}$
Σ	n × n covariance matrix of an n-dimensional random vector
Σ_A	covariance matrix of an actual state vector X_A
Σ_{REL}	covariance matrix of an estimated relative state vector (from onboard tracking)

$\sum_T$ MSFN tracking covariance matrix

$\sum_{\delta\bar{x}}$ n × n covariance matrix of $\delta\bar{x}$

$\sum_{\bar{\eta}}$ n × n covariance matrix of $\bar{\eta}$

σ standard deviation of a random variable with normal distribution

σ_{ij} element of the ith row and jth column of $\sum$

σ_{N_R} standard deviation for the noise on the range measurement

$\sigma_{N_{\dot{R}}}$ standard deviation for the noise on the range rate measurement

σ_{N_β} standard deviation for the noise on the shaft angle measurement

σ_{N_θ} standard deviation for the noise on the trunnion angle measurement

$\bar{\omega}_A$ actually turning rate vector for the thrust direction

$\bar{\omega}_D$ desired turning rate vector for the thrust direction

4.0 COORDINATE SYSTEMS

The coordinate system in which AGAST does most of its computations is called the basic reference frame. The AGAST basic reference frame corresponds to that of the CMC and the LM PGNCS. For earth missions, the reference coordinate system is the Besselian inertial frame; for moon reference, it is the selenocentric coordinate system. The AGS computer actually uses as its basic reference computing frame the definition of the PGNCS platform orientation at AGS-to-PGNCS alinement; but in AGAST, the AGS basic reference system is the same as that of the PGNCS. The following sections define some of the other coordinate systems used in AGAST.

4.1 Apollo Local Vertical Frame

The Apollo local vertical frame is defined for a given vehicle at a given time. Where X_{LV}, Y_{LV}, and Z_{LV} represent the axes of that coordinate system, the directions are defined as follows.

X_{LV} along the local horizontal in the direction of orbital motion

Y_{LV} along the negative of the angular momentum vector

Z_{LV} along the negative of the radius vector

4.2 The UVW Coordinate System

Another local vertical system used in AGAST is the UVW coordinate system. Where U, V, and W represent the axes of that coordinate system, the directions are defined as follows.

U along the radius vector

V along the local horizontal in the direction of orbital motion

W along the angluar momentum vector

4.3 Line-of-sight Coordinate System

The line-of-sight coordinate system is useful for some output purposes. A part of the onboard tracking simulation output is in that coordinate system. For the definition of the LOS system, the following notations are used: the indices i and j refer to either the LM or the CSM but not to both. Where X_{LOS}, Y_{LOS}, and Z_{LOS} represent the axes of the LOS coordinate system, the directions are defined as follows.

X_{LOS} along the line of sight from the ith vehicle to the jth vehicle

Y_{LOS} along the vector defined by the cross product of the line-of-sight vector with the ith vehicle's radius vector

Z_{LOS} along the vector which forms a right-handed system with X_{LOS} and Y_{LOS}

4.4 Nominal Platform Alinement Coordinate System

The CMC and PGNCS platform orientations denoted nominal are local vertical frames. The CMC nominal alinement system is the same as the Apollo local vertical defined in section 4.1. The PGNCS nominal platform alinement has a different definition. Where X_{NA}, Y_{NA}, and Z_{NA} represent the axes of the PGNCS nominal alinement coordinates, the directions are defined as follows.

X_{NA} along the positive radius vector

Y_{NA} along the negative angular momentum vector

Z_{NA} along the local horizontal in the direction of orbital motion

4.5 Preferred Platform Alinement Coordinate System

The inertial platforms of the CSM and the LM are usually put into the preferred maneuver orientation before a maneuver is executed. Where X_{PA}, Y_{PA}, and Z_{PA} represent the axes of the preferred platform coordinate system, the directions are defined as follows.

X_{PA} along the initial thrust direction

Y_{PA} along the vector formed by crossing the thrust vector into the radius vector

Z_{PA} along the vector which forms a right-handed system with X_{PA} and Y_{PA}

4.6 Landing Site Platform Alinement Coordinate System

During LM descent to the lunar surface and ascent into lunar orbit, the LM platform will be alined to the landing site alinement. That alinement is defined by the following definition of axes.

X_{LS} along the landing site radius vector

Y_{LS} along the direction which forms a right-handed system with X_{LS} and Z_{LS}

Z_{LS} along the vector formed by the cross product of the CSM angular momentum with the radius vector of the landing site

5.0 TRAJECTORY ERROR ANALYSIS TECHNIQUE

The following two sections describe the general Monte Carlo error analysis technique and its specific use in AGAST.

5.1 Statistical Trials Technique

Before a mission is flown, it is desirable to know the statistical characteristics of certain important trajectory parameters, for example, the covariance matrix of state vector uncertainties at some point in a rendezvous mission plan. The classical way to determine statistical information about an occurrence (or event) subject to random perturbation is to perform that event enough times experimentally to develop sufficient empirical data on its behavior. Obviously, that method is not appropriate for trajectory error analysis. Another method that can be used is that of statistical trials, in which the basic idea is substitution of a mathematical model for experimentation. That method requires a mathematical model of the occurrence under investigation (such as the flight of a spacecraft) and a model of the random perturbations that affect the occurrence. The computational procedure of the statistical trials method of error analysis it to perform the computations which model the occurrence (with the random perturbations) enough times to generate sufficient data to determine its statistical characteristics. Each time that those computations which model the event are made, a mathematical model of the perturbations introduces into the computations random values for the perturbations. There are standard formulations used to determine the number of times that a model must be cycled to produce a given confidence level in the validity of the computation of statistical characteristics based on the sample results (ref. 1).

Whenever the occurrence to be investigated is the trajectory of a spacecraft, it is desirable to determine the statistical characteristics of the trajectory when subject to specified perturbations with random distributions. For example, it may be desired to randomly perturb spacecraft engine performance and to determine the effects upon the trajectory. Each perturbed trajectory produced by the model is referred to as a sample actual trajectory. Whenever a mission is really flown, estimates of the actual trajectory are maintained in onboard and ground computers. A trajectory error analysis model also may simulate the computation of such estimates for each sample actual trajectory produced. An estimate of a sample actual trajectory which simulates the computations of an onboard or ground computer is referred to as a sample estimated trajectory.

5.2 AGAST Trajectory Simulation

The way in which the concept of statistical trials is used in AGAST to perform trajectory error analysis is described in this section. A nominal or reference trajectory is first produced by AGAST for both the LM and CSM; computations of the reference trajectory precedes the generation of perturbed Monte Carlo sample trajectories. To generate the nominal trajectory, the events of the input-defined mission plan are simulated with no navigation or guidance errors introduced into the computations.

The basic trajectory error analysis scheme of the program is to simulate for two vehicles an actual trajectory and to simulate estimated trajectories for the onboard and RTCC computers. The actual (environmental) state vectors are initialized as random dispersions of the initial reference state vectors. The estimated (onboard and RTCC) trajectories are dispersions of the actual trajectory. That is, the program maintains a true trajectory and onboard and RTCC estimates of that trajectory.

Discrepancies between the actual trajectory and the onboard estimates are assumed to occur because of error sources which include onboard integrator inaccuracies, measurement errors in the onboard sensors, engine performance discrepancies, and MSFN tracking uncertainties. A pseudorandom number generator is used in the introduction of random errors to the simulation of onboard and RTCC computations. The actual state vectors are propagated with an analytic ephemeris predictor which is assumed to be error free. The estimated state vectors are propagated with routines which simulate the onboard integrators.

After the program generates the reference trajectory, the computational sequence is cycled to produce (with error sources randomly sampled) several simulations of the mission profile. Each simulation consists of the initialization and maintenance of actual trajectories and, for each monitored computer, estimated trajectories. When a given mission plan is entered as input to the program, one of the input parameters is an integer N that represents the number of statistical trials (simulations) of the mission which the program user desires to have produced. The program will automatically cycle N times to generate N sets of actual and estimated trajectories for the mission plan defined by the input event packages. Because the error sources are randomly sampled, no trajectory in the set of N simulations will be duplicated. A summary of the trajectories produced by AGAST is as follows.

a. A reference mission trajectory X_R is produced for both vehicles.

b. For each Monte Carlo cycle, an actual mission trajectory X_A is computed for both vehicles.

c. For each Monte Carlo cycle, an estimated mission trajectory X_E is computed for both vehicles for each monitored computer.

The collection of actual and estimated trajectories produced is then processed to determine its statistical characteristics. Included in the statistical analysis are some of the burn parameters of the maneuvers. That is, after the required number of Monte Carlo simulations are produced, the statistical characteristics of the trajectory dispersions and the dispersions of maneuver parameters are computed. The statistics include maneuver ΔV means and standard deviations; covariance matrices that represent uncertainties in state vectors also are produced.

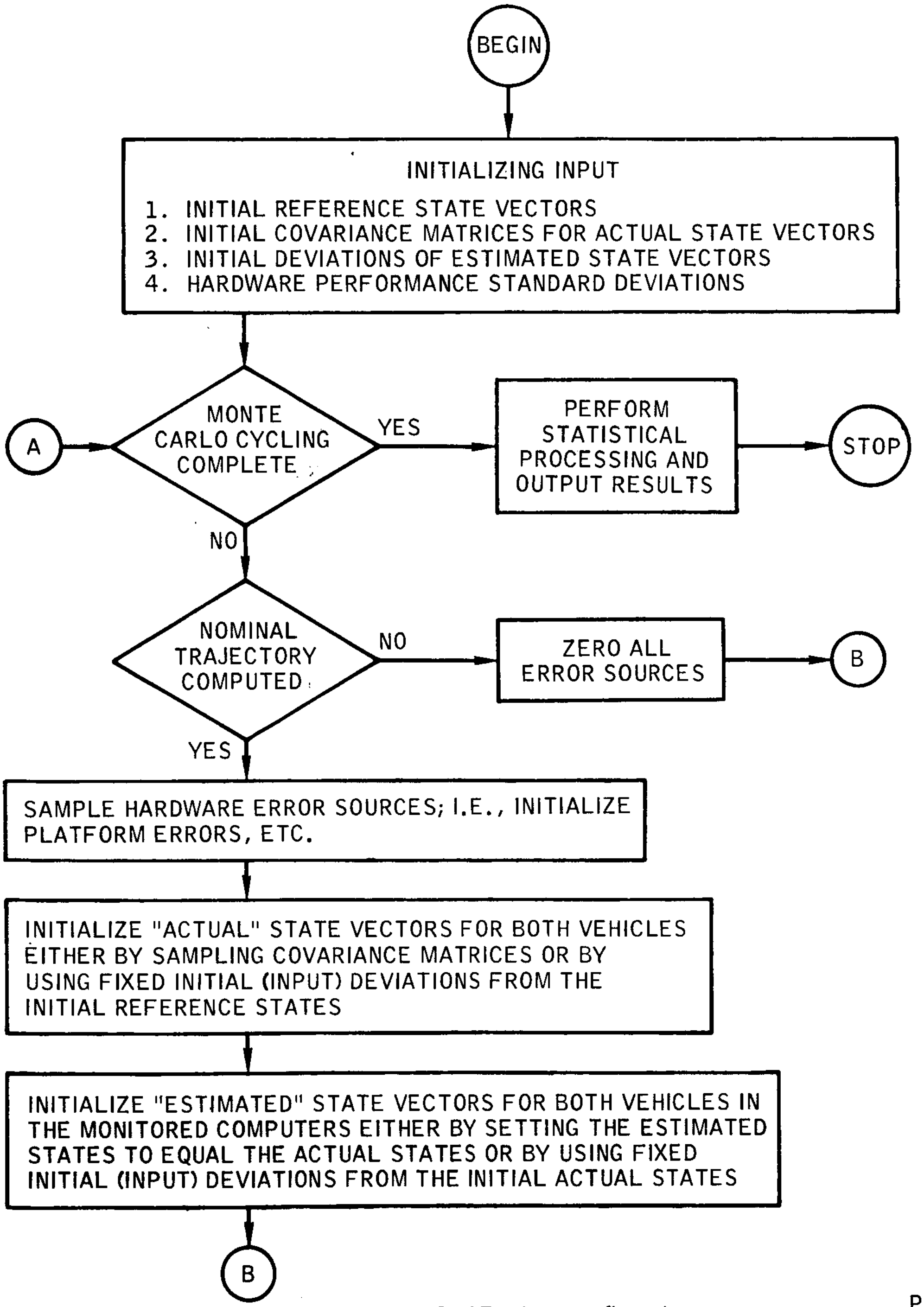

BEGIN
INITIALIZING INPUT
1. INITIAL REFERENCE STATE VECTORS
2. INITIAL COVARIANCE MATRICES FOR ACTUAL STATE VECTORS
3. INITIAL DEVIATIONS OF ESTIMATED STATE VECTORS
4. HARDWARE PERFORMANCE STANDARD DEVIATIONS
A
MONTE CARLO CYCLING COMPLETE
YES
PERFORM STATISTICAL PROCESSING AND OUTPUT RESULTS
STOP
NO
NOMINAL TRAJECTORY COMPUTED
NO
ZERO ALL ERROR SOURCES
B
YES
SAMPLE HARDWARE ERROR SOURCES; I.E., INITIALIZE PLATFORM ERRORS, ETC.
INITIALIZE "ACTUAL" STATE VECTORS FOR BOTH VEHICLES EITHER BY SAMPLING COVARIANCE MATRICES OR BY USING FIXED INITIAL (INPUT) DEVIATIONS FROM THE INITIAL REFERENCE STATES
INITIALIZE "ESTIMATED" STATE VECTORS FOR BOTH VEHICLES IN THE MONITORED COMPUTERS EITHER BY SETTING THE ESTIMATED STATES TO EQUAL THE ACTUAL STATES OR BY USING FIXED INITIAL (INPUT) DEVIATIONS FROM THE INITIAL ACTUAL STATES
B

Figure 1.- AGAST schematic flow chart.

Page 1 of 2

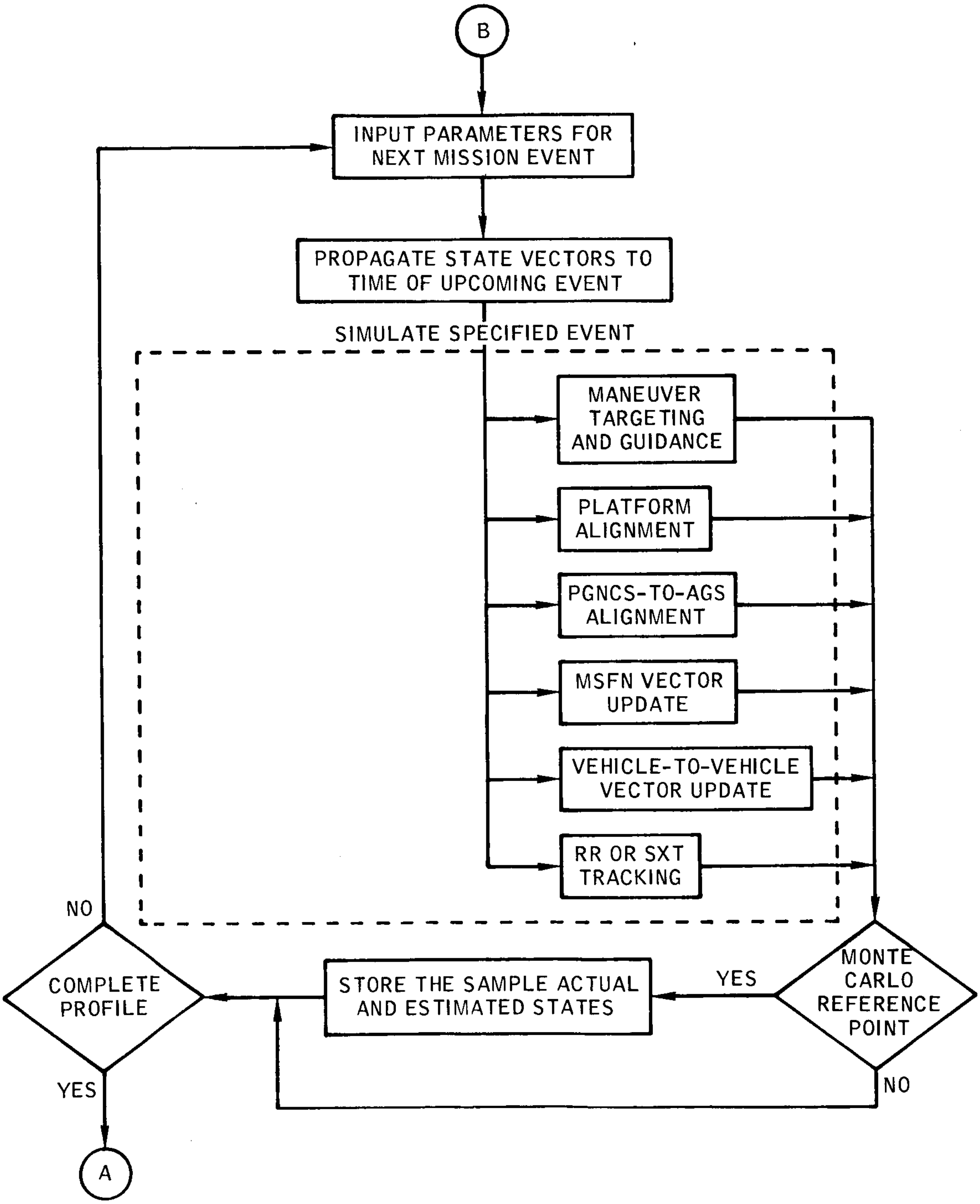

B
INPUT PARAMETERS FOR NEXT MISSION EVENT
PROPAGATE STATE VECTORS TO TIME OF UPCOMING EVENT
SIMULATE SPECIFIED EVENT
MANEUVER TARGETING AND GUIDANCE
PLATFORM ALIGNMENT
PGNCS-TO-AGS ALIGNMENT
MSFN VECTOR UPDATE
VEHICLE-TO-VEHICLE VECTOR UPDATE
RR OR SXT TRACKING
NO
COMPLETE PROFILE
YES
A
STORE THE SAMPLE ACTUAL AND ESTIMATED STATES
YES
MONTE CARLO REFERENCE POINT
NO

Figure 1.- Concluded.

Page 2 of 2

6.0 PROGRAM CAPABILITIES

The capabilities and options of AGAST are presented in this section. Subroutine names are capitalized. Further information is presented in the section on methods, in the individual routine descriptions, and in the user's manual (ref. 5). The reader is referred to section 3.2 of this document for definition of symbols used in the following discussion.

6.1 Computer Monitoring

The program user may have the program monitor any combination of CMC, PGNCS, AGS, and RTCC computers. When a computer is monitored, AGAST simulates the computations of that computer and maintains estimated state vectors for it.

An option exists in AGAST to have the AGS monitor a PGNCS controlled burn and vice versa. When one LM computer monitors a burn controlled by the other, the first computer accepts accelerometer sensed ΔV increments and accordingly integrates its estimate of the LM state vector.

6.2 Error Sources Modeled

Discrepancies arise between the reference and actual trajectories and the actual and estimated trajectories because of the following error sources.

a. Onboard integrator inaccuracies

b. MSFN tracking uncertainties

c. Platform misalinement and drift

d. Accelerometer misalinement, bias, and scale factors

e. Rendezvous radar noise and bias on range, range rate, shaft, and trunnion measurements

f. Sextant noise and bias on the angle measurements

g. VHF ranger noise and bias on range measurement

h. Engine thrust profile discrepancies

i. Thrust misalinement

The introduction of these error sources into the computations is described in the section on methods used and in the subroutine descriptions in the appendix.

6.3 Integrators and Central Body Options

The user may specify that either the earth potential model or lunar potential model be used. This capability permits error analyses to be performed for earth rendezvous missions and for lunar rendezvous missions.

The program maintains an actual trajectory, denoted by X_A, for both the LM and the CSM. Propagation of the actual trajectory during coasting flight is achieved by an analytic ephemeris predictor (AEG) and during powered flight by a modified average-g integrator (RUNGA). The perturbation models of the AEG and the powered flight integrator have more terms than the corresponding onboard integrators. Numerical values of the perturbation constants used in the reference integrators are periodically updated to correspond to the findings of the most recent potential studies.

The CMC and PGNCS estimates of the LM and CSM state vectors are propagated during coasting flight by the onboard coasting integrator (ENCKE). The onboard coasting integrator does not model atmospheric drag in near-earth orbits. Powered flight state vector integration is accomplished in those computers by the onboard average-g integrator (AVEG). A nonperturbed central body potential model is used in the AGS computer for both coasting and powered flight integration of AGS estimated state vectors. The AGS coasting integrator and powered flight integrator as modeled in AGAST are, respectively, AGSKEP and AGSAG.

6.4 Program Initialization

Initialization of AGAST is accomplished by specification of the following.

a. Computers to be monitored (i.e., CMC, PGNCS, AGS, RTCC)

b. Appropriate central body (earth or moon)

c. Desired number of Monte Carlo trajectory simulations

d. Initial reference state vectors for the CSM and LM, denoted by X_R

e. Engine characteristics

f. Numerical hardware error models

Several options exist in the initialization process. Numerical models of the Apollo hardware performance are programed into AGAST, but the program user may specify changes to any of those models. The prestored constants are listed in the appendix as defined in reference 6 and include the following.

a. Nominal engine thrust profiles

b. Nominal lunar landmarks for optical tracking

c. One-sigma engine performance discrepancies

d. One-sigma platform and accelerometer errors

e. One-sigma RR, SXT, VHF, and SCT measurement errors

Each time that the program is cycled by the Monte Carlo logic, all the random Gaussian errors are reinitialized; the model is resampled.

The initial dispersions of the actual state vectors X_A are also reinitialized for each Monte Carlo cycle. The initial actual state vectors may be initialized as follows.

a. For either vehicle, X_A may be defined as a random dispersion δX_A from X_R computed by sampling an input covariance matrix of actual state vector uncertainties, denoted by $\sum_A$.

b. For either vehicle, X_A may be defined by input to be a specified deviation δX_A from X_R. In this way, the user may, if he wishes, have $X_A = X_R$.

The initial estimated state vectors X_E for any monitored computer are always initialized to be a specified (input) deviation from X_A. There exists no option to obtain the initial estimated states by the process of sampling a tracking covariance matrix because that capability exists as a mission event option (MSFN update). That is, if a user wishes to model MSFN tracking uncertainties, he may specify the first mission event to be a MSFN update and enter as input a matrix of tracking uncertainties, $\sum_T$. This method is the one most often used.

The covariance matrix $\sum_A$ represents uncertainties of the dispersions of the initial actual state vector from the initial nominal X_R. For instance, $\sum_A$ may represent the uncertainties of launching the LM into a nominal lunar orbit from the lunar surface.

The initialization of X_A (for one vehicle) from a covariance matrix of uncertainties about X_R is presented in figure 2. Also presented is the subsequent modeling of the MSFN determination of the estimated state X_E. The sequential process followed by AGAST which is schematized in figure 2 is as follows.

a. X_R is defined by input

b. $\sum_A$ is defined by input

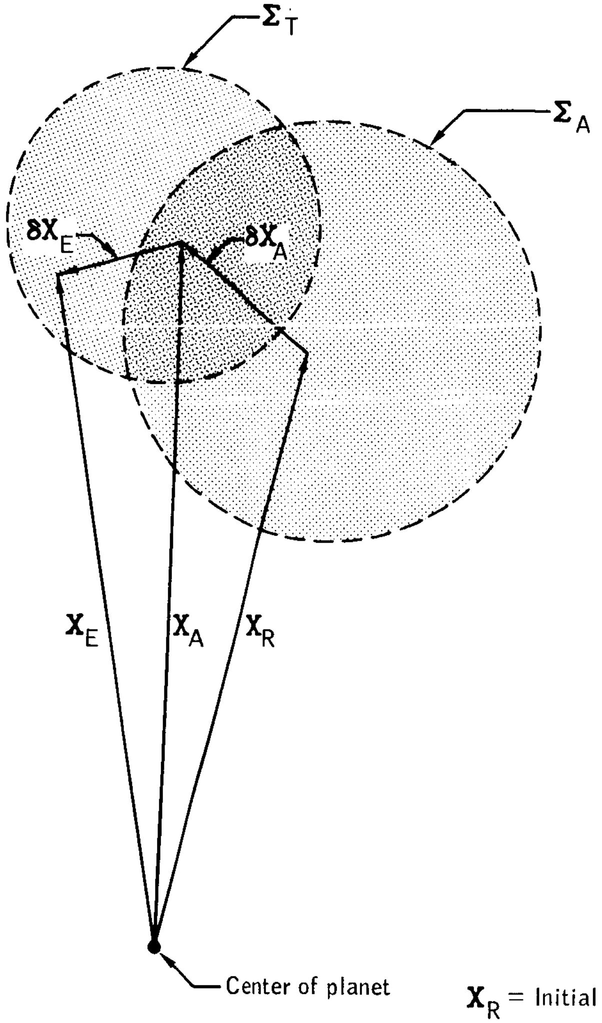
c. δX_A is computed as a random sampling of $\sum_A$

d. $X_A = X_R + \delta X_A$

e. Input specifies a MSFN update mission event: $\sum_T$ is input.

f. δX_E is computed as a random sampling of $\sum_T$.

g. $X_E = X_A + \delta X_E$



X_R = Initial input

$X_A = X_R + \delta X_A$

$X_E = X_A + \delta X_E$

Figure 2.- Initialization of AGAST state vectors.

6.5 Maneuver Computations

The discussion of maneuver computations in AGAST is divided into a section on maneuver simulation and a section on special targeting computations for comparative studies. The associated logic is presented in more detail in the flow charts of AGAST subroutines MANEUV and TARGET.

6.5.1 Maneuver simulation.- A variety of options exist in AGAST for the computation of maneuver targeting and for simulation of guidance. One of those options is that the numerical thrust model of any engine may be changed at any point in the simulation of the mission plan. The program user may use one of the following options on any maneuver event which he schedules.

a. Targeting is computed, but the ΔV is not applied to the state vector.

b. The targeted ΔV is impulsively applied to the state vector (with no maneuver execution errors modeled).

c. The burn is controlled by onboard guidance equations

1. With system errors not modeled

2. With system errors modeled

The following lists present all the maneuver events which may be simulated in AGAST.

a. Target ΔV: updates the CSM(LM) state in the PGNCS (CMC) by impulsive simulation of a maneuver which has been performed by the CSM(LM)

b. External ΔV guidance

c. SOI: stable orbit initiation

d. SOM: stable orbit midcourse

e. SOR: stable orbit rendezvous

f. CSI: coelliptic sequence initiation

g. CDH: constant delta height

h. TPI: terminal phase initiation

i. TPM: terminal phase midcourse

j. Direct intercept

k. Direct transfer to input target

l. TPF: braking

m. NCC: corrective combination

n. NSR: coelliptization

o. N_{C1} DKI phasing maneuver

p. N_H height maneuver

q. Staging

r. Plane change

s. DOI: descent orbit injection

t. Descent: powered descent to the lunar surface

u. Ascent: powered ascent into lunar orbit

v. Abort: abort from a powered descent to a powered ascent into lunar orbit

The following list is a summary of the capabilities of the AGAST models of the different computers.

a. The RTCC model can generate targeting for any maneuver in the above list.

b. The PGNCS can handle all the computations indicated by the above list except for DOI and the ground (RTCC) computed maneuver sequences that involve NCC, NSR, N_{C1}, and N_H.

c. The CMC can perform all the maneuver computations except DOI, lunar descent, lunar ascent, and ground computed maneuvers NCC, NSR, N_{C1}, and N_H.

d. AGAST currently has AGS capability only for CSI, CDH, TPI, TPM, and external ΔV guidance, ascent guidance, and abort guidance.

6.5.2 Special maneuver targeting computations.- Often it is desirable to determine for a given maneuver the targeting parameters which the different computers will produce. Such information is useful in analyzing and setting up computational procedures for the mission plan under study.

For a given maneuver, AGAST has the capability to generate targeting based on the estimated states of any or all of the four computers (with the computer restrictions noted in section 6.5.1) and also to compute the targeting based on the actual state vectors. Subsequent to those computations, AGAST will produce the following.

a. Targeting discrepancies between the different computers

b. Targeting discrepancies between the solution based on the actual state vectors and the solutions based on the estimated states of the computers

The burn simulation is controlled by the targeting computed for the controlling computer.

Because the different computers will produce different maneuver times for some maneuvers, the targeting based on the actual states is computed for each of those different times. The program user may command AGAST to compute the following solutions for any given maneuver in a mission.

a. The CMC ΔV solution at CMC predicted maneuver time: CMC/CMC

b. The PGNCS ΔV solution at PGNCS predicted maneuver time: PGNCS/PGNCS

c. The AGS ΔV solution at AGS predicted maneuver time: AGS/AGS

d. The RTCC ΔV solution at RTCC predicted maneuver time: RTCC/RTCC

e. The ΔV solution based on the actual state vectors but computed at the CMC predicted maneuver time: ACT/CMC

f. The ΔV solution based on the actual state vectors but computed at the PGNCS predicted maneuver time: ACT/PGNCS

g. The ΔV solution based on the actual state vectors but computed at the AGS predicted maneuver time: ACT/AGS

h. The ΔV solution based on the actual state vectors and computed at the predicted maneuver time based also on the actual state vectors: ACT/ACT

Coincident with the computation of special maneuver targeting is the generation of the resultant target differences. Depending upon which targeting solutions are computed, some or all of the following ΔV target differences are formed by AGAST, and their statistical characteristics are produced.

a. CMC/CMC - ACT/CMC

b. PGNCS/PGNCS - ACT/PGNCS

c. AGS/AGS - ACT/AGS

d. RTCC/RTCC - ACT/ACT

e. PGNCS/PGNCS - CMC/CMC

f. PGNCS/PGNCS - AGS/AGS

g. PGNCS/PGNCS - RTCC/RTCC

h. RTCC/RTCC - CMC/CMC

i. RTCC/RTCC - AGS/AGS

j. CMC/CMC - AGS/AGS

6.6 Platform Alinements

Platform alinements for the CMC and for the PGNCS are simulated by computation of a platform orientation matrix REFSMMAT for each computer. The platform orientation matrix is the rotation matrix from the basic reference coordinate system to the coordinate system of the inertial platform. The REFSMMAT matrix is used in many onboard computations; for instance, during a burn the sensed ΔV increments are measured in the coordinate system of the inertial platform but must be transformed into the basic reference coordinate system which the guidance uses. In AGAST, the computation of the REFSMMAT at a given time in the mission is an implicit definition of a platform alinement. The REFSMMAT may be computed for the following types of alinement.

a. Nominal alinement is defined to be the alining of the platform gyros to a local vertical coordinate system defined at the alinement time.

b. Preferred maneuver alinement is based on the desired thrust direction of a given maneuver.

c. Lunar surface alinement is used during lunar landing and lunar launch and is based on the radius vector of the landing site.

The specific REFSMMAT formulations associated with those three types of alinements are detailed in the flow charts of the AGAST subroutines ALIGN, PMATRX, PALIGN, and LSALGN.

Platform errors are reinitialized with each alinement. The errors are modeled as initial platform misalinement at the alinement time and platform gyro drift after alinement. These errors are represented in AGAST as a coordinate system rotation error in the definition of REFSMMAT. The associated rotation error matrix is introduced into the guidance computations and in the incorporation of onboard tracking measurements.

6.7 AGS-to-PGNCS Alinement

One procedure by which the AGS computer is initialized is as follows.

a. Define the AGS reference coordinate system to be that of the PGNCS inertial platform.

b. Load into the AGS the PGNCS estimated state vectors rotated into the AGS reference coordinate system.

This procedure is called AGS-to-PGNCS alinement. The AGS may be initialized in AGAST in this manner as a mission event. The required operations are modeled in subroutine ALIGN, but AGAST AGS computations are made in the basic reference frame to facilitate comparison of AGS with the other computers. However, AGS platform errors which are caused by PGNCS misalinement at the AGS-to-PGNCS alinement time are accounted for.

6.8 State Vector Updates

Several options exist in AGAST for the modeling of updates of the estimated state vectors maintained in the monitored computers. Updates during a mission may be accomplished by transferral of state vector information from one computer to another, by MSFN orbit determinations, and by the use of onboard tracking devices.

6.8.1 Vehicle-to-vehicle transferral.- The estimated state vectors of one computer may be used to replace those of one of the other computers. This mission event models the procedure by which the CMC update of the PGNCS (or vice versa) is accomplished by the voicing of state vector information between the two vehicles.

6.8.2 MSFN updates.- MSFN orbit determination and the subsequent update of the estimated states in the computers are modeled in AGAST by the method of randomly sampling a covariance matrix of MSFN tracking uncertainties. Whenever the program user specifies that a mission event is a MSFN update of one of the computers, he must also enter as input to AGAST a matrix of state vector tracking uncertainties $\sum_T$. That matrix implicitly correlates the MSFN CSM state vector uncertainties with the MSFN LM state vector uncertainties. Both estimated states in the specified computer are updated by sampling $\sum_T$ and determining new estimated states at the mission event time.

The random sampling method is discussed in section 7.1. The update logic which models the MSFN is to be found in AGAST subroutine REPLAC.

6.8.3 Onboard tracking.- Onboard tracking is modeled in AGAST in two ways.

a. Random sampling of an input onboard tracking covariance matrix

b. Explicit modeling of the onboard Kalman filters and the associated navigation equations

Updates for both CSM and LM tracking devices can be simulated.

A rendezvous radar (RR) is attached to the LM. The RR measures the range and range rate between the LM and CSM by the pulsing of signals to a beacon on the CSM and the sensing of the pulse returned by that beacon. Also measured are the radar disk shaft and trunnion angles which relate the line-of-sight direction and the LM body attitude. The PGNCS is interfaced with the RR; and when the PGNCS is in the tracking incorporation mode, it samples the output of the RR approximately once a minute. The PGNCS uses a modified Kalman filter to incorporate the tracking parameters into the estimate of the state vector of either the CSM or the LM.

The AGS computer may accept range and range rate information manually keyed into the computer by a crewman. The crewman may read the RR tracking data from a display. The AGS RR filter incorporates the tracking data into the LM state vector estimate.

Two optical tracking devices are part of the equipment of the CSM. One of the instruments is a sextant (SXT), and the other is called a scanning telescope (SCT). Also attached to the CSM is a VHF ranger, a device used to measure the range between the CSM and LM. The SXT and VHF tracking data are passed through the same filter. In this document, the symbol SXT will refer to both SXT and VHF tracking.

These CSM optical instruments may be directed by a crewman toward either the LM or a lunar landmark. When the operator depresses the MARK button on a CSM console, the time and measured line-of-sight angles are stored in the CMC. When the VHF range measuring device is being used, the measured relative range is automatically stored in the CMC, and that computer picks up the VHF output approximately once a minute. Through the use of a modified Kalman filter, the measured relative information is incorporated into the CMC estimate of the CSM state vector if the CMC is in the lunar landmark tracking mode or either the CSM or LM estimated state if the CMC is in the LM tracking mode.

6.8.3.1 Covariance matrix modeling of onboard tracking: Except for lunar landmark tracking, the effect of the state vector updates accomplished with the onboard tracking devices and filters is to reduce the errors in the estimated relative state between the two vehicles. The onboard computer logic which handles such updates is based on the assumption that the estimate of one of the vehicles' state vector is error free. Both RR tracking of the CSM and SXT tracking of the LM can be modeled in AGAST by the sampling of a corresponding covariance matrix of tracking uncertainties $\sum_{REL}$ for the relative state vector. In that way, a random estimate of the relative state vector is formed; and, in turn, an estimate of one of the vehicles' state vector is made. That method of modeling entails no explicit use of onboard filters. The input matrix $\sum_{REL}$ must be generated external to AGAST. The relevant AGAST logic is in subroutine REPLAC, and the details of the formulation are presented in section 7.4.

6.8.3.2 **Explicit modeling of onboard tracking:** The Kalman filters for CMC incorporation of SXT and SCT data, for PGNCS incorporation of RR data, and for AGS incorporation of RR data are programed into AGAST. The program user may specify any number of tracking events for a mission. The program will simulate the operation of the tracking devices by the production of tracking data for each mark followed by the performance of the computations of the filter and subsequent updates of one of the estimated state vectors. The AGAST subroutines UPDATV and AGSRR are the program entry points for tracking logic and formulation.

7.0 METHODS AND FORMULATIONS

Detailed discussions of some of the more important modeling methods and formulations used in AGAST are presented in this section. Much of the formulation used in AGAST originated in the referenced documents and is adequately discussed in those reports.

7.1 Random Sampling Technique

The modeling method of statistical trials is based upon the technique of generation of random dispersions for certain parameters considered to have random distributions. All AGAST parameters with random dispersions are assumed to have normal (Gaussian) distributions.

Several references have been made in the preceding text to the process of sampling error sources. For example, each time that the program is cycled, a new set of initial state vector dispersions are generated by randomly sampling a covariance matrix of the uncertainties of the state vector. An explanation of that process will be presented in this section.

To validly perform the generation of dispersions for a given random variable, the following are required.

a. An input set that defines the statistical properties of the variable, that is, the variance for a scalar variable or the covariance matrix for a vector variable

b. A random number generator

There exist standard computer routines for the generation of random numbers. AGAST uses a UNIVAC 1108 math-pack routine RANDN which produces random arrays for which the components have a mean of zero and a variance of 1. The use of RANDN is discussed in the description of AGAST subroutine RANNO. In the following discussion, assume that the normalized random vector $\bar{\eta}$ is generated by a random number generator such as RANDN.

Consider the random vector $\bar{\eta}$ to be a normalization of the random vector $\delta\bar{x}$. That is, assume $\delta\bar{x} = C\bar{\eta}$ where C is an $n \times n$ constant matrix that relates $\delta\bar{x}$ to $\bar{\eta}$. The problem is to determine the multiplier C.

By definition,

$$\sum_{\bar{\eta}} = E\,[\bar{\eta} \cdot \bar{\eta}^T] = I \qquad (1)$$

where E is the statistical expectation operator.

Also, note that

$$\sum_{\delta\bar{x}} = E\ [\delta\bar{x} \cdot (\delta\bar{x})^T] \qquad (2)$$

If $\delta\bar{x}$ is set equal to $C\bar{\eta}$ and is substituted into equation (2), equation (3) results.

$$\begin{aligned}\sum_{\delta\bar{x}} &= E\ [C\bar{\eta} \cdot (C\bar{\eta})^T] \\ &= E\ [C\bar{\eta} \cdot \bar{\eta}^T C^T] \\ &= C \cdot E\ [\bar{\eta} \cdot \bar{\eta}^T] \cdot C^T \\ &= C \cdot C^T \end{aligned} \qquad (3)$$

Therefore, the matrix C is a factor in a decomposition of $\sum_{\delta\bar{x}}$. However, the decomposition CC^T of a square matrix is not unique. **Assume** that C will be lower left triangular and therefore unique.

Let σ_{ij} and c_{ij} represent, respectively, the element of $\sum_{\delta\bar{x}}$ and C in the ith row and jth column. The matrix can be written as follows.

$$C \cdot C^T = \begin{bmatrix} C_{11}^2 & C_{11}C_{21} & \cdots & C_{11}C_{n1} \\ C_{21}C_{11} & C_{21}^2 + C_{22}^2 & \cdots & C_{21}C_{n1} + C_{22}C_{n2} \\ \vdots & \vdots & \vdots & \vdots \\ C_{n1}C_{11} & C_{n1}C_{21} + C_{n1}C_{22} & \cdots & \sum_{k=1}^{n} C_{nk}^2 \end{bmatrix}$$

When the elements of $\sum_{\delta\bar{x}}$ and CC^T are equated and when C_{ij} is solved for, equation (4) results.

$$C_{ij} = 0$$

where $j > i$.

$$C_{11} = \sqrt{\sigma_{11}}$$

$$C_{ii} = \sqrt{\sigma_{ii} - \sum_{k=1}^{i-1} C_{ik}}$$

$$C_{ij} = \frac{\sigma_{ji} - \sum_{k=1}^{i-1} C_{jk}C_{ik}}{C_{ii}} \qquad (4)$$

where $j < i$.

Compute the matrix C when $\sum_{\delta\bar{x}}$ is given as input and by use of the algorithm indicated by the above formulation. Each time that it is desired to sample the matrix $\sum_{\delta\bar{x}}$, it will only be necessary to generate an n-dimensional array of random numbers symbolized by $\bar{\eta}$ and then set the random dispersion $\delta\bar{x}$ to be $C\bar{\eta}$.

Note that when the variable $\bar{x}$ is a scalar the multiplier C also reduces to a scalar. The sampling technique described above is used in AGAST to produce dispersions for all variables $\bar{x}$ which are subject to random variation $\delta\bar{x}$. Variables which are dispersed randomly include state vectors, platform alinements, engine parameters, and onboard tracking measurements. Sampling of covariance matrices is performed in subroutine SAMPLE.

7.2 Maneuver Execution Error Model

Nonnominal execution of maneuvers is modeled in AGAST in three ways.

a. Nonnominal engine performance (thrust levels, etc.)

b. Introduction of platform and accelerometer errors into the computation of guidance commands and the simulation of sensed $\Delta\bar{V}$

c. At the end of a burn, approximation of the burn error caused by an initial thrust pointing discrepancy

Nonnominal engine performance is simulated by dispersion of the thrust profile from the nominal. Thrust dispersions are generated in each thrust phase either as a percentage of the nominal thrust magnitude for that phase or by randomly sampling the standard deviation of the thrust magnitude for that phase.

7.2.1 Platform and accelerometer error models for CMC and LGC maneuver simulation.- The guidance logic and equations used by subroutine MANEUV (fig. B-55) generate either a desired thrust direction $\bar{u}_{TD}$ or a desired thrust vector turning rate $\bar{\omega}_D$ in basic reference coordinates. Those commands are transformed into IMU coordinates and then into body referenced coordinates. The digital autopilot (DAP) accepts the steering commands in body referenced coordinates and attempts to execute them. Because the inertial platform's alinement is not perfect, the transformation matrix REFSMMAT is in error by the amount of rotation of the platform from its desired orientation. Therefore, the transformation of basic reference frame guidance commands with REFSMMAT does not produce the correct transformation to the true orientation of the platform so that the steering commands which the DAP receives do not have the correct orientation. The error in the steering commands is modeled in AGAST by determination of the true orientation of the platform and subsequent determination of the basic reference orientation of the guidance command executed by the DAP.

AGAST does not contain a model of the DAP. When the guidance command is $\bar{u}_{TD}$, the burn simulator (RUNGA) generates a turning rate vector similar to $\bar{\omega}_D$ which rotates the thrust direction at a constant rate from the old thrust direction to the new $\bar{u}_{TD}$. Whether the thrust command is generated as $\bar{u}_{TD}$ or $\bar{\omega}_D$, the turning rate is limited to be less than or equal to 10 deg/sec. Both $\bar{u}_{TD}$ and $\bar{\omega}_D$ are rotated to reflect the REFSMMAT error before they are processed by RUNGA.

The object of the platform error model is to apply the effects of the platform errors to an output of the guidance equations in such a way that the burn simulation will reflect the true reaction of the steering system. Both the guidance equations and the burn simulator compute in basic reference coordinates. Therefore, the error model formulation must convert the guidance command into IMU coordinates, apply the platform errors, and then reconvert to basic reference coordinates. The platform errors consist of initial misalinement in each of the three gyros at the alinement time and of gyro drift since alinement time. At alinement time, the platform errors are determined either by setting them to input specifications or by sampling the numerical model of platform statistics. The platform errors are the three rotation angles that define platform misalinement about the three axes of the REFSMMAT coordinate system. Let [P(t)] represent the platform error matrix at the time t, and let [T] represent the REFSMMAT matrix. Then if $\bar{u}_{TD}$ or $\bar{\omega}_D$ is produced as output from the guidance equations, the values $\bar{u}_{TA}$ and $\bar{\omega}_A$ are entered as input to the burn simulator, where

$$\bar{u}_{TA} = [T]^{-1} \cdot [P(t)] \cdot [T] \cdot \bar{u}_{TD} \qquad (5)$$

$$\bar{\omega}_A = [T]^{-1} \cdot [P(t)] \cdot [T] \cdot \bar{\omega}_D \qquad (6)$$

The burn simulator accepts the guidance command and performs the burn computations which include the following.

a. The $\Delta\bar{V}$ added during the guidance

b. New thrust direction (in basic reference frame) at the end of the guidance cycle

c. Integration of the actual trajectory to the end of the guidance cycle

The $\Delta\bar{V}$ output from the burn simulator is considered to be the actual and is symbolized by $\Delta\bar{V}_A$; the value of $\bar{u}_{TA}$ is updated in the burn simulator and represents the new actual thrust direction.

In actual flight, the accelerometers attached to the platform sense the burn and feed to the computer the measured incremental velocity. The measured $\Delta\bar{V}$ is transformed from IMU coordinates to basic reference coordinates and is used in the onboard average-g state vector integrator to integrate the estimated state vector forward one guidance cycle. The AGAST modeling of accelerometer and platform errors on the measured $\Delta\bar{V}$ fed to the onboard computer is as follows.

$$\Delta\bar{V}_E = [T]^{-1} \left\{ [S] \cdot [B] \cdot [P(t)] \cdot [T] \cdot \Delta\bar{V}_A + \Delta\bar{V}_B \right\} \qquad (7)$$

Integration of X_E is performed in AVEG, which accepts $\Delta\bar{V}_E$ as an input.

The new thrust direction $\bar{u}_{TA}$ which is produced as output from the burn simulator is also perturbed with platform errors before being passed on to the guidance equation.

$$u_T = [T]^{-1} \cdot [P(t)] \cdot [T] \cdot \bar{u}_{TA} \tag{8}$$

where $\bar{u}_T$ is the AGAST representation of the onboard estimate of the new thrust direction.

7.2.2 Platform and accelerometer error models for AGS maneuver simulation.- The AGS error models differ in formulation from those of the PGNCS because the AGS reference frame is defined to be the orientation of the PGNCS platform at the time of AGS-to-PGNCS alinement. Therefore, PGNCS platform errors at the time of AGS alinement affect the AGS guidance computations. For an AGS controlled burn, the thrust vector sent to the burn simulator in AGAST is

$$\bar{u}_{TA} = [T]^{-1} \cdot [A(t)]^{-1} \cdot [P_A(t)] \cdot [A(t)] \cdot [P_T] \cdot [T] \cdot \bar{u}_{TD} \tag{9}$$

where $\bar{u}_{TD}$ is generated in basic reference coordinates and $[A(t)]$ relates the AGS strapped down platform to the AGS reference computing frame (not used in AGAST). The burn simulator routine RUNGA outputs a new $\bar{u}_{TA}$ at the end of the 2-second guidance cycle, and the new estimated thrust vector passed on to the AGS guidance equations is

$$\bar{u}_T = [T]^{-1} \cdot [A(t)]^{-1} \cdot [P_A(t)] \cdot [A(t)] \cdot [P_T] \cdot [T] \cdot \bar{u}_{TA} \tag{10}$$

The actual incremental velocity of a guidance cycle $\Delta\bar{V}_A$ is output from RUNGA. Formulation in AGAST for computation of the incremental velocity sensed by the AGS accelerometer over a guidance cycle is as follows.

$$\Delta\bar{V}_M = [S_A] \cdot [B_A] \cdot [P_A(t)] \cdot [A(t)] \cdot [P_T] \cdot [T] \cdot \Delta\bar{V}_A \tag{11}$$

$$\Delta\bar{V}_E = [T]^{-1} \cdot [A(t)]^{-1} \cdot \Delta\bar{V}_M \tag{12}$$

The value $\Delta\bar{V}_E$ is used by the AGS average-g integrator AGSAG to advance the AGS estimated LM state one guidance cycle.

7.2.3 Pointing error model and burn trim.- The effects of thrust pointing errors, accelerometer quantization, or manual maneuver cutoff errors are accounted for in AGAST subroutine TRIM by the introduction of random velocity perturbations into the states at burn cutoff. Thrust pointing error is caused by insufficient knowledge in the onboard guidance computer of the spacecraft center of gravity. During attempts to correct for the center of gravity error, the guidance computer introduces cross-axis errors into the burn. Cross-axis errors are in the plane normal to the thrust direction. The pointing error velocity is simulated in AGAST when a normal distribution is assumed for the error and a random sampling is used based upon a standard deviation value taken from the cross-axis velocity curves in reference 6.

RCS burns are manually controlled, and for such maneuvers, the vehicle usually is placed in an attitude which places the desired $\Delta\bar{V}$ along one of the body thrusting axes. Because no steering is involved in RCS burns, pointing errors are not considered. The RCS burn error considered in AGAST is based on the crewman's inability to release the throttle at exactly the proper time. A random velocity error is computed in the direction of the thrust to account for RCS cutoff error.

Quantization error results because the accelerometers measure $\Delta\bar{V}$ applied during a burn as an integral number of PIPA pulses so that some partial pulse is applied but not included in the accelerometer readout. AGAST accounts for this effect by addition to the actual state of random $\Delta\bar{V}$ which is uniformly distributed over the quantization interval of the accelerometer.

The use in AGAST of the pointing error $\Delta\bar{V}_P$ and the manual cutoff error $\Delta\bar{V}_{CO}$ depends upon the trim option desired. If no trim is desired, then $\Delta\bar{V}_P$ and $\Delta\bar{V}_{CO}$ must be added to the actual and estimated states to account for the errors that would result in a burn with no trim. For a burn which is to be trimmed, AGAST does not model pointing errors or manual cutoff errors; that is, it is assumed that those errors will be taken out by trimming the residuals. Trims are simulated by impulsive applications to both the actual and estimated states of part or all of the $\bar{V}_G$ residual that results from guidance errors. If the program user specifies the perfect trim option, then the trim $\Delta\bar{V}$ applied is equal to the $\bar{V}_G$ residual; otherwise, the residual in each axis is trimmed to a preset value.

7.3 Onboard Tracking Error Models

The onboard tracking devices are subject to bias and noise errors in their measurements. Numerical models of the bias and noise are programed into AGAST. Each time that the program starts another Monte Carlo cycle of a given mission plan, all biases are fixed by randomly sampling the numerical models. For example, there is a standard deviation value stored in AGAST for the bias in the RR measurement of the dish shaft angle. AGAST simulates that bias by randomly sampling that standard deviation each new cycle of the program. Other biases are similarly modeled.

For each mark of an onboard tracking schedule, the tracking logic of AGAST simulates the measured parameters as follows.

Measured = Actual + Bias + Noise

The measured relative range, range rate, shaft and trunnion angles are computed as follows.

$$R_M = R_A + \delta R + (\sigma_{N_R}) \cdot \eta_1 \qquad (13)$$

$$\dot{R}_M = \dot{R}_A + \delta \dot{R} + (\sigma_{N_{\dot{R}}}) \cdot \eta_2 \qquad (14)$$

$$\beta_M = \beta_A + \delta\beta + (\sigma_{N_\beta}) \cdot \eta_3 \qquad (15)$$

$$\theta_M = \theta_A + \delta\theta + (\sigma_{N_\theta}) \cdot \eta_4 \qquad (16)$$

where the δ-values are the biases, the σ_N-values are the standard deviations of the noise, and the η_i are normalized random numbers which are generated for each mark. Similarly, the optical and VHF data measured onboard the CSM are simulated by computation of the actual parameters and addition of bias and noise values. For the SXT and SCT, the AGAST error model takes the form of rotation bias and noise in the unit vector along the measured line of sight. The VHF relative range measurement is formulated exactly as that for the RR. The standard deviation values for RR and VHF measurement bias and noise are functions of the range (ref. 6). These standard deviation functions are modeled in the programed tracking logic.

7.4 Covariance Matrix of Onboard Tracking Uncertainties

As previously explained, AGAST may model the effects of onboard vehicle-to-vehicle tracking by sampling an input matrix of relative state vector uncertainties. The assumptions and logic formulations of that option are presented in this section.

The vehicle-to-vehicle onboard tracking measurements are of relative state vector parameters. For the RR, these parameters are range, range rate, and the radar dish orientation angles shaft and trunnion. The SXT produces a unit vector directed from the CSM to the LM, and the VHF device produces range data. The measurement incorporation algorithm updates the estimated state vector of one of the vehicles in such a manner that the error in the measurement parameters tends to converge to zero. This method of updating does not account for the fact that the estimated state vector of the other vehicle may also be in error.

The computation sequence in AGAST which uses a covariance matrix of relative state vector uncertainties to model onboard tracking is as follows.

a. Compute ΔX_A in the local vertical coordinate system of $X_{A,J}$.

b. Define $\Delta X_E^* = \Delta X_A$.

c. ΔX_E^* is assumed to be a vector centered at $X_{E,J}$ and is rotated from the local vertical of $X_{E,J}$ to the basic reference coordinate system.

d. Define $X_{E,I}^* = X_{E,J} + \Delta X_E^*$.

e. Compute δx_{REL} by sampling $\sum_{REL}$.

f. Rotate δx_{REL} from the local vertical coordinate system of $X_{E,I}^*$ to the basic reference frame.

g. Define $X_{E,I} = X_{E,I}^* + \delta X_{REL}$

When the dispersion δX_{REL} is zero, the new estimated relative state vector is ΔX_E^*, which produces the same relative range, range rate, and local vertical line-of-sight angles as does ΔX_A. Note that ΔX_E^* is the best relative state vector that the onboard tracking may produce;

that is, the relative state directed from $X_{E,J}$ corresponds in a local sense to ΔX_A.

The computer logic of this option is formulated in the description of AGAST subroutine REPLAC. A schematic of the geometrical relationships of the above computational sequence is presented in figure 3. In this figure, the angle β is the actual line-of-sight angle from the local horizontal.

7.5 Computation of Sample Statistics

When a program user defines a mission plan to AGAST, he may specify that certain events be reference points. Such reference points are those events of the mission plan for which the program operator desires to produce sample state vector statistics and sample ΔV statistics. For each cycle of the program by the Monte Carlo logic, sample actual state vectors and sample estimated state vectors of the reference points are saved on magnetic tape. If the reference point is a maneuver event, targeting and burn parameters are also stored.

After the requested number of statistical trials of a trajectory has been computed, AGAST then reads the sample state vectors and maneuver parameters from the tape and performs statistical processing of that sample data. For each reference point, the AGAST statistical processing logic computes the array of dispersions of the sample actual state vectors from the corresponding nominal state vector at that point. Also computed is the array of dispersions of sample estimated state vectors from the corresponding sample actual state vectors. These arrays of dispersions are processed by the logic of subroutine PROCES. The following statistics are produced.

a. Sample means and sample standard deviations for the total ΔV required to fly the mission plan

b. Cumulative distribution of dispersions to the nominal total ΔV required

c. Sample means and sample standard deviations for the components of the $\overline{\Delta V}$ targeting solutions for the maneuver events

d. Sample means and sample covariance matrices for state vector dispersions

The specific logic used and parametric output are discussed in detail in the description of subroutine PROCES. The formulation by which

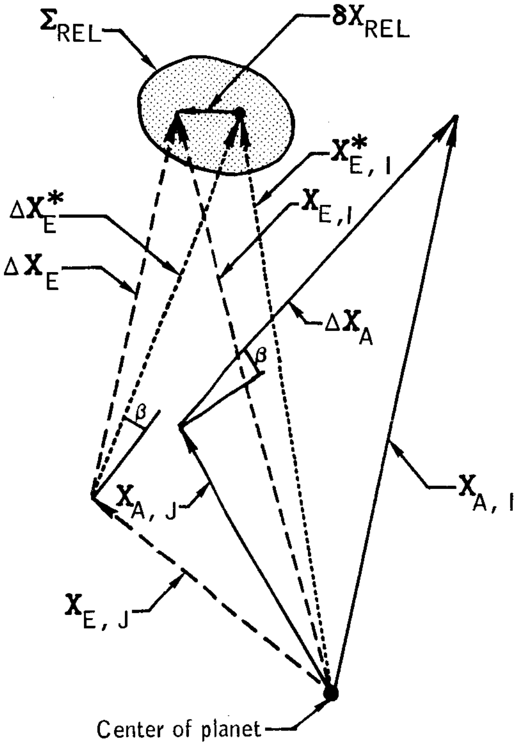


Figure 3.- Using relative state vector covariance matrix to simulate onboard tracking.

AGAST computes sample statistics is presented below and is based on derivations in reference 7.

If the following symbols are defined as shown,

X	a random vector
$\left\{X_i\right\}_{i=1}^{N}$	an array of random samples of X
$\bar{\mu}$	mean of X
σ	standard deviation of $\lvert X\rvert$
Σ	covariance matrix of X

then the formulation used in AGAST is

$$\bar{\mu} = \frac{1}{N} \sum_{i=1}^{N} X_i \tag{17}$$

$$\sigma = \sqrt{\frac{1}{N-1}\left[\sum_{i=1}^{N} X_i^2 - \frac{1}{N}\left(\sum_{i=1}^{N} X_i\right)^2\right]} \tag{18}$$

$$\Sigma = \frac{1}{N-1}\left\{\sum_{i=1}^{N} X_i \; X_i^T - \frac{1}{N}\left(\sum_{i=1}^{N} X_i\right)\cdot\left(\sum_{i=1}^{N} X_i\right)^T\right\} \tag{19}$$

APPENDIX A

DESCRIPTIONS OF THE SUBROUTINES OF AGAST

APPENDIX A

DESCRIPTIONS OF THE SUBROUTINES OF AGAST[a]

AEG (Analytic Ephemeris Generator)

Purpose: To propagate analytically state vectors to a specified time

Called by: AFLPN, AGSRR, AMAIN, BUTTON, CFAEG5, COEDH, COMDKI, DELTAV, ENCKE, FILTER, INPUT, JOCDH, LOPC, MANEUV, ONR, PCA, PMISS, PROPAG, REPLAC, RTCCSI, TARGET, TAUAL, TCONE, TILL, TIMAL, TIMFAL, TPASS, UPDATV

Input: ELE, CART, SPHERE, T_{SEC}, K_{REF}, L_1, L_2, KORIN, KOROUT, I_{SET}, I_{WRAP}, K_1

Output: ELE, CART, SPHERE

Comments: Subroutine AEG in AGAST is a driver which advances state vectors by use of the earth analytic ephemeris generator (ref. 8) or the lunar analytic ephemeris generator (ref. 9). The K_{REF} flag indicates whether the earth or lunar potential is to be used.

AFLPN (Ascent Flight Plan Routine)

Purpose: To compute the desired lift-off time, the ascent targets, and the inertial position and velocity of the actual and estimated landing sites at lift-off time

Called by: AMAIN, INPUT2

Input: H_A, H_P, Θ_{INS}, L_{SA}, L_{SE}, Σ_{LSA}, Σ_{LSE}, $\delta\chi_{LSA}$, $\delta\chi_{LSE}$, ϕ_D, T_{EVENT}

Output: X_A, X_E, T_{LO}, R_{DD}, V_R, V_H

[a] Symbols used in this appendix are defined in appendix C.

AGS (Abort Guidance System)

Purpose: To simulate the onboard AGS guidance routines

Called by: MANEUV, TARGET

Input: See maneuver input defined in reference 5

Output: Either maneuver targets, if used by TARGET for targeting, or thrust direction and magnitude, if used by MANEUV for guidance

Comments: See sections 6.3.7 and 7.19 to 7.30 of reference 4

AGSAG

Purpose: To simulate 2-second integration step of the AGS average-g method of propagation of LM state (either powered or coasting flight)

Called by: AGSKEP, MANEUV

Input: $\bar{r}(t)$, $\bar{v}(t)$, $\bar{\Delta}v$

Output: $\bar{r}(t+2)$, $\bar{v}(t+2)$

Comments: See sections 6.3.4 and 7.13 of reference 4

AGSKEP

Purpose: To propagate AGS LM and CSM state vectors during coasting flight

Called by: FILTER, MANEUV, PROPAG, TARGET

Input: $\bar{r}(t)$, $v(t)$, $\bar{r}_c(t)$, $\bar{v}_c(t)$, Δt

Output: $\bar{r}(t+\Delta t)$, $\bar{v}(t+\Delta t)$, $\bar{r}_c(t+\Delta t)$, $\bar{v}_c\ (t + \Delta t)$

AGSRR

Purpose: To simulate AGS rendezvous radar tracking and updating of AGS LM state vector

Called by: AMAIN

Comments: Present program uses G-mission tracking schedule

ALDENS

Purpose: To compute atmospheric density for use in subroutine DRAG

Called by: DRAG

Comments: See reference 8

ALIGN

Purpose: To compute the REFSMMAT [T] for the desired alinement and initialize the platform errors associated with the alinement

Called by: AMAIN

Input: T_{EVENT}, L_{EN}, I_{COM}, I_{VEH}, I_{PE}, alinement option flag

Output: [T]

Comments: See references 2 and 3 for a discussion of the different types of IMU alinement

AMAIN

Purpose: This routine is the main driver for the AGAST program

Comments: The logic of AMAIN calls the imput routines, the routines which initialize the state vectors for the monitored computers, and the routines which simulate the mission events. The Monte Carlo loop is controlled from the main program. State vectors and other pertinent data for the reference points are written on tape unit 13 by AMAIN. When the Monte Carlo simulation is complete, AMAIN calls up the statistical processor PROCES to generate sample means, standard deviations, and covariance matrices for state vectors and maneuver parameters of the reference points.

AMATRX

Purpose: To compute [A], the rotation matrix from the Besselian reference system to Greenwich true of date coordinates

Called by: INPUT

Input: B_{YEAR}, B_{MONTH}, B_{DAY}

Output: [A]

Comments: The subroutine first computes the Julian date of the nearest Besselian year as described in reference 10; then subroutines MANGLE, NUTE, and PRESS are used to compute the data needed to obtain [A].

ANOM

Purpose: To compute mean anomaly from true anomaly

Called by: AEG, CONVRT

Input: e, θ

Output: M

APSIDE

Purpose: To compute the apocynthion and pericynthion radii and the eccentricity of an orbit defined by input

Called by: CFP

Input: $\bar{R}_1$, $\bar{V}_1$

Output: R_A, R_P, e

Comments: See section 5.5.14 of reference 3

ASCENT

Purpose: On each guidance cycle, to update the thrust direction which will insert the LM into the desired orbit. This program is used for ascent from the lunar surface or for aborts from powered descent.

Called by: MANEUV

Input: X_E, ΔT_{TO}, X_A, A_{TM}, τ, V_E, R_{DD}, Y_{DD}, $\bar{X}_{DD}$, I_{FPC}, I_{FLUP}

Output: $\bar{U}_T$, T_{GO}, $\Delta\bar{V}_G$

Comments: This subroutine is an engineering simulation of the logic described in section 5.3.5 of reference 3

AVEG (Onboard Powered Flight Integrator)

Purpose: To propagate estimated state during powered flight

Called by: MANEUV

Input: $\bar{R}(T)$, $\bar{V}(T)$, ΔT

Output: $\bar{R}(T+\Delta T)$, $\bar{V}(T+\Delta T)$

Comments: See section 5.3.2 of reference 3

AVGOBL

Purpose: To compute the gravity vectors used by subroutine AVEG

Called by: AVEG, MANEUV

Input: $\bar{R}$

Output: $\bar{g}$, $\bar{g}_b$

Comments: See section 5.3.2 of reference 3

BUTTON

Purpose: To control logic for the targeting computations for the N_{CC} and N_{SR} maneuvers

Called by: TARGET

Comments: See reference 11 for a discussion of the logic of BUTTON

CATIME

Purpose: Driver logic for the computation of a table of times of CSM closest approach to the landmarks to be tracked

Called by: UPDATV

Input: N_{LM}, $\{\phi_i, \lambda_i\}_{i=1}^{N_{LM}}$

Output: N_{TOTAL}, T_{TRK} - Table

CBA

Purpose: To compute the accelerometer misalinement matrix for the indicated computer

Called by: ALIGN

Input: I_{COM}, accelerometer ε-array

Output: [B] for the indicated computer

CDFUNC

Purpose: To compute a cumulative distribution function table for a sample array of a random variable (such as total fuel consumed)

Called by: PROCES

Input: K, N, $\{X_j\}_{j=1}^{N}$

Output: $\{p_i\}_{i=1}^{K}$

Comments: The output array is computed as follows.

$$p_i = \text{probability that } X < X_{min} + i \cdot \Delta X$$

CFAEG5

Purpose: Logic required to compute conic fit two-impulse rendezvous solutions based on the analytic ephemeris generator and a Lambert targeting routine

Called by: BUTTON, TURNUP

Comments: See reference 9

CFP (Concentric Flight Plane Routine)

Purpose: To compute the CSI and CDH maneuvers by use of the onboard targeting logic

Called by: TARGET

Input for CSI: T_{CSI}, T_{TPI}, E_L, N_{CDH}, I_{VEH}; for CDH: T_{CDH}, T_{TPI}, E_L, I_{VEH}

Output: $\bar{\Delta} V_{CSI}$ (LV), $\bar{\Delta} V_{CDH}$ (LV), T_{CDH}

Comments: See programs P32 and P33 described in section 5.4.2 of reference 3

COE

Purpose: This routine consists of the coelliptic orbit equations which compute the orbital parameters of an orbit that is coelliptic to an input orbit and separated from the input orbit by ΔH_{CDH}

Called by: SOR

Imput: ΔH_{CDH}, a_J, $V_{V,J}$, $\bar{R}_J$, $\bar{V}_J$

Output: $\bar{R}_I$, $\bar{V}_I$

COEDH

Purpose: To compute the targeting for the N_{SR} maneuver: $\Delta\bar{V}$ required and coelliptic orbital elements

Called by: RTCCSI, TARGET

Comments: See reference 13

COMDKI

Purpose: To use modified CSI/CDH targeting logic to compute DKI maneuvers for the following four sequences: N_{C1}-N_{SR}; N_{C1}-N_H-N_{SR}; N_H-N_{C1}-N_{SR}; N_H-N_{SR}.

Called by: TARGET

Input: K_{TARG}, I_{TEV}, TH_{CL}, ΔT_{CL}, N_1, N_2, N_3, T_{N1}, T_{N2}, T_{N3}, L_{EN}, $X_{E,I}$, $X_{E,J}$, $X_{A,I}$, $X_{A,J}$

Output: $\Delta\bar{V}_1$, $\Delta\bar{V}_2$, $\Delta\bar{V}_3$, T_{N1}, T_{N2}, T_{N3}

Comments: A general discussion of the four maneuver sequences targeted by COMDKI is presented in reference 14

COMMPT

Purpose: To load and display the mission plan table

Called by: AMAIN

Input: T_{EX}, $\overline{\Delta V}_{EX}$, SAVMAN array

Output: Mission plan table display

CONIC

Purpose: To propagate state vectors by use of the universal variable formulation of Kepler's equation

Called by: CFP, DESCNT, ENCKE, MIDCRS, PRETPI, TRITPM

Comments: Conic is equivalent to the Kepler subroutine of section 5.5.5 of reference 2

CONVRT

Purpose: To convert input vectors to a form suitable for use by the AEG

Called by: INPUT

Input: X, I_{SET}, I_{CON}

Output: ELE, CART, SPHERE

COVTAB

Purpose: Initially to store input covariance matrices and subsequently reset those matrices when commanded by the logic of the program

Called by: INCOV, REPLAC

Input: I_{TYPE}, K_D, I_{COV}, I_{ROT}, Σ

Output: Σ

CPEM

Purpose: To compute the platform misalinement rotation matrix for the indicated computer

Called by: AGSRR, ALIGN, MANEUV, PERRO, UPDATV

Input: I_C, T_A, $\theta_{x,o}$, $\theta_{y,o}$, $\theta_{z,o}$, $\dot{\theta}_x$, $\dot{\theta}_y$, $\dot{\theta}_z$

Output: $[P(t)]$

CROSP (Onboard Cross Product Steering Routine)

Purpose: To compute the burn T_{GO} and compute the commanded turning rate for each guidance cycle

Called by: MANEUV

Input: $\bar{V}_G$, $\bar{U}_{TA}$ K_{STEER}, I_{VEH}, $\overline{\Delta V}$, $\Delta T_{TAIL\text{-}OFF}$

Output: $\bar{U}_{TD}$, $\bar{W}_C$, T_{GO}, K_{STEER}

Comments: Section 5.3.3 of reference 3 contains a discussion of cross-product guidance

CROSS

Purpose: To cross two input vectors $\bar{a}$ and $\bar{b}$ and output the product $\bar{c}$.

DELTAV

Purpose: To perform impulsive simulation of a maneuver

Called by: MANEUV

Input: I_{PASS}, T_{DV}, $X_{E,I}$, $\Delta\bar{V}_{LV}$, L_{EN}, I_{COM}, $X_{A,I}$

Output: $X_{E,I}$, $X_{A,I}$

Comments: This routine is used both to simulate a burn impulsively and to perform the target ΔV operations of the onboard computers

DESCNT

Purpose: To update the thrust magnitude and thrust direction every 2 seconds which will guide the LM to a desired landing site. This is the powered descent guidance program.

Called by: MANEUV

Input: X_E, W_T, I_{phase}, $[T]$, $\bar{R}_{LS}$

Output: $\bar{U}_T$, $\bar{T}_{GO}$, PDI, F_c, I_{sp}

Comment: This subroutine is an engineering simulation of part of the logic in section 5.3.4 of reference 3. It does not include a simulation of the landing radar or of any body oriented displays.

DINCP1

Purpose: To compute the navigational Kalman filter parameters and compute the correction to be made to the state vector

Called by: DMESIP

Input: J_D, $\bar{\alpha}^2$, $\bar{b}_1$, $\bar{b}_2$, $\bar{b}_3$, $[W]$

Output: a, [Z], [Ω], [δX]

Comments: This subroutine corresponds to INCORP1 as described in section 5.2.3 of reference 3.
In the DINCP1 flow chart of appendix B, the W-matrix is partitioned as follows.

$$[W] = \left(\begin{array}{c|c|c} W_1 & W_2 & W_3 \\ \hline W_4 & W_5 & W_6 \\ \hline W_7 & W_8 & W_9 \end{array}\right)$$

DINCP2

Purpose: To update the W-matrix

Called by: DMESIP

Input: $\bar{\alpha}^2$, J_D, [Z], [W]

Output: [W]

Comments: This subroutine corresponds to INCORP2 as described in section 5.2.3 of reference 3.
The W-matrix of the DINCP2 flow chart of appendix B is partitioned as follows.

$$[W]= \begin{pmatrix} \bar{W}_1 & \bar{W}_2 & \cdots & \bar{W}_9 \\ \bar{W}_{10} & \bar{W}_{11} & \cdots & \bar{W}_{18} \\ \bar{W}_{19} & \bar{W}_{20} & \cdots & \bar{W}_{27} \end{pmatrix}$$

DMESIP

Purpose: Utility routine which calls DINCIP1 and DINCP2 to compute the correction to the state vector and subsequently update the state vector

Called by: ONR, RNR, RR

Input: LM_{UP}, $X_{E,1}$, $X_{E,2}$, $\bar{\delta}_1$, $\bar{\delta}_2$, $\bar{\nu}_1$, $\bar{\nu}_2$, $\delta\beta_E$, $\delta\theta_E$

Output: $X_{E,1}$, $X_{E,2}$, $\bar{\delta}_1$, $\bar{\delta}_2$, $\bar{\nu}_1$, $\bar{\nu}_2$, $\delta\beta_E$, $\delta\theta_E$

DMTMPY

Purpose: To compute the double precision product [C] of the double precisioned m by n matrix [A] and the double precisioned n by p matrix [B]

Called by: AMATRX

Input: [A], [B], m, n, p

Output: [C]

Comments: [C] is computed by

$$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

DOI (Descent Orbit Insertion Routine)

Purpose: To compute the descent orbit insertion maneuver

Called by: TARGET

Input: T_{EVENT}, $\bar{X}_{LS}$

Output: T_{EX}, $\overline{\Delta V}_{EX}$

Comments: See subroutine LLTPR of reference 15

DOT

Purpose: To compute the dot product of two vectors

DRAG

Purpose: To compute the effects of atmospheric drag on a state vector propagated through a given time interval by the earth AEG

Called by: ESAEG

Comments: See reference 8

ELPRE

Purpose: To propagate AGS state vectors in AGS guidance routine by use of Keplerian motion

Called by: AGS, AGSKEP

Comments: See figure 7.33 of reference 4

ENCKE (Onboard Coasting Integrator)

Purpose: To propagate estimated state vectors during coasting flight

Called by: CFP, DOI, INITV, LAMAIM, MANEUV, MIDCRS, ONR, PRETPI, PROPAG, QRDTPI, RNR, RR, SOR, SPLIT, TARGET, UPDATV

Input: ELE, CART, SPHERE, T_{SEC}, L_1, L_2, I_{SET}, KORIN, KOROUT, K_1

Output: ELE, CART, SPHERE

Comments: Subroutine ENCKE is the AGAST simulation of the onboard coasting integrator for the CMC and the LGC (section 5.2.2 of ref. 3)

EPERT

Purpose: To compute the perturbing acceleration caused by planetary oblateness

Called by: RUNGA

Input: x, y, z, μ, J_2, J_3, J_4, J_{22}, J_{31}, K_{REF}

Output: x, y, z

ERRSXT

Purpose: To compute the error matrix for the unit vector that represents the sextant alinement to a landmark and then rotate the true unit vector through the error matrix to obtain an errored unit vector

Called by: UPDATV

Input: $\bar{U}_{SXT}$, δ_{SXT_x}, δ_{SXT_y}, δ_{SXT_z}, σ_{SXT_x}, σ_{SXT_y}, σ_{SXT_z}

Output: $\bar{U}$

ESAEG (Earth AEG)

Purpose: To propagate a state vector in earth orbit

Called by: AEG

Comments: See reference 8

ETOXYZ

Purpose: To convert Keplerian orbit elements to Cartesian elements

Called by: AEG, ENCKE

Input: a, e, i, g h, M

Output: $\bar{R}$, $\bar{V}$

EXDV

Purpose: To compute the initial velocity-to-be-gained vector for an external ΔV guided maneuver

Called by: MANEUV

Input: $\bar{R}$, $\bar{V}$, $\overline{\Delta V}_{LV}$, F, W_E

Output: $\bar{V}_G$

Comments: See section 5.3.3.3.1 of reference 3 for a discussion of external ΔV prethrust computations

FILTER

Purpose: To simulate the onboard AGS rendezvous radar filter which updates the LM state vector

Called by: AGSRR

Comments: See figures 7.16 through 7.19 of reference 4

GEOMET (Geometric Parameters Routine)

Purpose: To compute various parameters defined from two input vectors

Called by: LAMBRT, PARAM

Input: $\bar{\eta}_1$, $\bar{\eta}_2$

Output: η_1, η_2 $\bar{U}_{\eta_1}$, $\bar{U}_{\eta_2}$, cos ϕ, sin ϕ, $\bar{N}$, $\bar{U}_N$

Comments: This subroutine is documented in section 5.5.10 of reference 3

GMATRX

Purpose: To compute the product [C] of the m by n matrix [A] and the n by p matrix [B]

Called by: (General utility routine)

Input: [A], [B], m, n, p

Output: [C]

Comments: [C] is computed by

$$C_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

GMTIME

Purpose: To convert days, hours, minutes, and seconds to seconds and vice versa

Called by: COMMPT, DESCNT, INPUT1, OUTPUT

GUESS

Purpose: To compute an optimum solution for the $N_{CC} - N_{SR}$ sequence

Called by: TURNUP

Comments: The iterative technique used in GUESS is documented in reference 16. There is no detailed flow chart for this routine.

INCOV

Purpose: Input routine for those matrices whose dimension is a multiple of six

Called by: INPUT1, INPUT2

Input: K_D

INITV (Initial Velocity Routine)

Purpose: To compute the velocity required to intercept a specified point in space at a specified time

Called by: LAMAIM, MIDCRS, PRETPI, SOR, VTBG

Input: $\bar{R}$, $\bar{V}$, $\bar{R}_T$, T_2, E, N_1

Output: $\bar{V}_T$, $\bar{R}_{T_2}$

Comments: This subroutine is discussed in section 5.5.11 of reference 3

INPUT

Purpose: Routine for the reading of the base date and reference state(s) into the program and the computation of the A-matrix or initialization of the ephemeris tape

Called by: INPUT1

Input: K_{REF}, L_1, L_2

Output: T_R, $\bar{X}_R$, Base Date

INPUT1

Purpose: Routine for the reading of program initialization data

Called by: AMAIN

Comments: See reference 5 for a detailed description of the initialization input

INPUT2

Purpose: Routine for the reading of the event package input for each event to be performed

Called by: AMAIN

Comments: See reference 5 for a detailed discussion of the event package input

INRTCC

Purpose:. To read the RTCC vector package

Called by: INPUT

Input: I_{VEH}

Output: $\bar{X}_R$, T_R

ITRATE

Purpose: To determine the zero of a monotonically increasing function or the minimum of a function whose first derivative is single valued in the desired range

Called by: CONIC, LAMBRT

Comments: See section 5.5.10 and figure 5.10-3 of reference 2

JDATE

Purpose: To compute the Julian date corresponding to the base date

Called by: PRESS

Input: B_{YEAR}, B_{MONTH}, B_{DAY}

Output: T_{JC}, T_{JD}

JOCDH

Purpose: To compute the CDH time slip required to hold TPI time fixed

Called by: TARGET

Input: K_{NSR}, T_{NSR}, T_{TPI}, ε_D

Output: K_{NSR}, T_{SAVE}

JPLEPH

Purpose: To compute the lunar libration matrix [PNL] and the positions of the sun and moon with respect to the earth

Called by: PWVCON, TAPEUP, TCONE, TILL, TRANCO

Comments: See reference 17

KEP (Kepler Equation Routine)

Purpose: To compute the transfer time in a Keplerian orbit for a given value of the universal variable χ

Called by: CONIC, LAMBRT, TIMEP

Input: C_1, C_2, χ, ζ, $\bar{R}$

Output: T_{21}, $C(\zeta)$, $S(\zeta)$

Comments: See reference 2, pages 5.5-42 and 5.5-47

LAMAIM

Purpose: To compute the initial $\bar{V}_G$ for a Lambert guided maneuver

Called by: MANEUV

Input: T_1, T_2, T_{IG}, $\bar{R}_T$, $\bar{R}$, $\bar{V}$

Output: $\bar{V}_G$

Comments: See reference 2, pages 5.3-23 to 5.5-26

LAMB

Purpose: To solve for the two-body initial velocity given the initial and final position vectors and a transfer time between the two

Called by: CFAEG5

Comments: LAMB is essentially equivalent to the LAMBERT subroutine of section 5.5.6 of reference 2 with the exception that multiple revolution transfer trajectory capability has been added.

LAMBRT

Purpose: To simulate the onboard Lambert program which computes the targeting for a two-impulse rendezvous solution

Called by: INITV

Comments: This subroutine is discussed fully in section 5.5.6 of reference 3

LOPC (CSM Lunar Orbit Plane Change Routine)

Purpose: To compute a maneuver to place the landing site in the CSM orbital plane at lift-off time

Called by: TARGET

Input: T_{EVENT}, $\bar{X}_{LS}$

Output: T_{EX}, $\overline{\Delta V}_{EX}$

Comments: See reference 18.

LSAEG (Lunar AEG)

Purpose: To propagate analytically the state vector of a vehicle in lunar orbit

Called by: AEG

Comments: Two versions of the LSAEG are used in AGAST. See reference 9 for the triaxial version and reference 19 for the generalized version.

LSALGN (Lunar Surface and Landing Site Alinement Routine)

Purpose: To compute the REFSMMAT [T] for a lunar surface alinement

Called by: ALIGN

Input: $\bar{R}_{LS}$, $\bar{H}$

Output: [T]

Comments: See reference 2, page 5.6-54

LVLHDX

Purpose: One route converts a state vector increment from LV coordinates to BR coordinates and adds the increment to the input state vector; the other route converts a state vector increment from BR to LV coordinates

Called by: REPLAC, SAMPLE

Input: $\bar{R}_o$, $\bar{V}_o$, $\Delta\bar{R}$, $\Delta\bar{V}$

Output: $\bar{R}$, $\bar{V}$

MANEUV

Purpose: To perform powered flight maneuvers

Called by: AMAIN

Input: See reference 5 for maneuver event input

Output: Actual and estimated state vectors after the maneuver

Comments: See section 5.3 of references 2 and 3

MANGLE

Purpose: To calculate various parameters relating to the orbits of the sun and moon which are used by subroutines AMATRX, NUTE AND PRESS

Called by: AMATRX

Comments: See reference 20

MATRIX

Purpose: To multiply a three dimensioned vector $\bar{B}$ by a 3 by 3 matrix [A]

Called by: (General utility routine)

Input: [A], $\bar{B}$

Output: $\bar{C}$

MDOT

Purpose: To compute the velocity of a point on the moon caused by lunar rotation

Called by: TRANCO

Comments: See reference 21

MIDCRS

Purpose: To compute the terminal phase midcourse maneuver and the direct intercept maneuver

Called by: TARGET

Input: T, T_2, active vehicle number

Output: $\bar{\Delta}\bar{V}_{EX}$

Comments: See program P35 of reference 2 or 3 (section 5.4.25)

MTRP

Purpose: To transpose an m by n matrix [A]

Called by: (General utility routine)

Input: [A], m, n

Output: $[A]^T$

NEWT

Purpose: To perform a fifth-order Newton interpolation on the data from the JPL ephemeris

Called by: JPLEPH

Comments: See reference 17

NUTE

Purpose: To compute the earth nutation terms needed in subroutine AMATRX

Called by: AMATRX

Comments: See reference 20

OBS

Purpose: To produce tracking data that simulates the observations of the RR and SXT; to incorporate the tracking equipment error models into the computations of measurements

Called by: UPDATV

Input: $\bar{R}_{REL}$, $\bar{V}_{REL}$, $[T_A]$, [SMNB], σ_{N_R}, $\sigma_{N_{\dot{R}}}$, σ_{N_β}, σ_{N_θ}, R_{BIAS}, δR, $\delta\dot{R}$, $\delta\beta$, $\delta\theta$, $\sigma_{N_{VHF}}$, VHF_{BIAS}, δR_{VHF}, δ_{SXT_x}, δ_{SXT_y}, δ_{SXT_z}, σ_{SXT_x}, σ_{SXT_y}, σ_{SXT_z}

Output: R_A, $\dot{R}_A$, β_A, θ_A, R_M, $\dot{R}_M$, β_M, θ_M, $\bar{U}_{REL}$, $\bar{U}_{SXT}$

ONR

Purpose: To accept simulated optics data and perform navigational updates to the CSM vector and the landmark radius vector

Called by: UPDATV

Input: X_E, I_{known}, L, K_{RNDFG}, N, $\bar{U}_{NB}$, [T], [NBSM], ϕ, λ, h_{LS}, $[W_{LRV}]$, $[W_{UNL}]$, $\bar{X}_{KL}$, VAR_{IMU}, VAR_{SCT}, VAR_{INT}, [W]

Output: X_E, [W]

Comments: Subroutine ONR is an engineering simulation of the logic described in section 5.2.5.2 of reference 2.
The W-Matrix of the ONR flowchart of appendix B is partitioned as follows.

$$[W] = \left(\begin{array}{c|c|c} W_1 & W_2 & W_3 \\ \hline W_4 & W_5 & W_6 \\ \hline W_7 & W_8 & W_9 \end{array}\right)$$

OPS

Purpose: To compute orbital parameters r, α, n, c, s for use by subroutine ELPRE

Called by: AGS, AGSKEP

Comments: See figure 7.33 of reference 4

ORD

Purpose: To order the table T_{TRK} in an ascending sequence

Called by: CATIME

Input: N_{TOTAL}, T_{TRK}-Table, K_{TRK}-Table

Output: T_{TRK}-Table, K_{TRK}-Table

OUTPUT

Purpose: Display routine for the orbital parameters of the reference state vector

Called by: INPUT1

Input: Reference state vector

Output: Printed display

PALIGN (Preferred Alinement Routine)

Purpose: To calculate the desired alinement for powered flight maneuvers

Called by: ALIGN, TRIM, ULLDIR

Input: $\bar{R}$, $\bar{V}$, $\bar{U}_{TD}$

Output: [T]

Comments: See reference 2, pages 5.6-53

PARAM

Purpose: To compute various orbital parameters for use in other routines

Called by: APSIDE, TIMEP

Input: $\bar{R}$, $\bar{V}$

Output: Cot γ, C_3, α_N, P_N

Comments: See reference 2, pages 5.5-43

PCA

Purpose: To compute the time and distance of closest approach between two vehicles

Called by: MANEUV

Input: vehicle state vectors

Output: T_{CA}, D_{CA}

PERRO

Purpose: To simulate the effects of platform and accelerometer errors on the $\Delta\bar{V}$ sensed by the systems

Called by: MANEUV

Input: L, I_{COM}, $\Delta\bar{V}_A$, $[B_C]$, $[B_P]$, $[B_A]$, REFSMMAT, [A(t)], scale factors, accelerometer read biases

Output: $\Delta\bar{V}_E$, [P(t)]

PMATRX

Purpose: To compute the rotation matrix from basic reference coordinates to the vehicle platform for various platform alinements

Called by: AFLPN, ALIGN, COMDKI, DELTAV, LOPC, LVLHDX, MANEUV, REPLAC, RNRPRT, SAMDIF, TPITPM

Input: $\bar{R}$, $\bar{V}$ option flag

Output: [T]

PMISS

Purpose: To advance the LM and CSM AEG states to T_{TPI} and to compute the phase angle discrepancy

Called by: COMDKI

Input: ε_D, T_{TPI}

Output: $\Delta\theta$

Comments: The desired phase angle at TPI is defined as that phase angle which produces an elevation angle ε_D. The phase angle discrepancy $\Delta\theta$ is then the difference between the desired phase angle θ_D and the phase angle θ_{TPI}.

PRESS

Purpose: To compute the earth precession term required by AMATRX

Called by: AMATRX

Comments: See reference 20

PRETPI

Purpose: To compute the terminal phase initiation maneuver

Called by: CFP, TARGET

Input: T_{TPI}, E_L, ωt, active vehicle number

Output: T_{TPI}, $\bar{\Delta}\bar{V}_{EX}$

Comments: See program P34 of reference 2, pages 5.4-22 to 5.4-27

PROCES

Purpose: To read the Monte Carlo data on the output tape and process those data to generate sample means, standard deviations, and covariance matrices

Called by: AMAIN, XMAIN

Input: I_{SAVE}, N_{SAM}; for each reference point, the following parameters are read from the data tape (maneuver parameters are not read for nonmaneuver events)

a. Reference state vectors and time
b. Reference maneuver ΔV burned
c. Sample state vectors and times
d. Sample maneuver $\Delta\bar{V}$ targeted
e. Sample maneuver ΔV burned
f. Sample maneuver burn time ΔT_B
g. Sample maneuver residual ΔV
h. Sample maneuver trimmed ΔV
i. The XTRA array
j. The SAVMAN array

Output: Statistics are printed

PROPAG

Purpose: To propagate actual and estimated state vectors to a given time

Called by: AMAIN, DELTAV, SPLIT, TARGET, UPDATV

Input: I_{COM}, I_{NOMT}, I_{PASS}, L_{EC}, L_{EN}, T_{EVENT}, T_{START}, T_{STOP}, I_{WMATRX}

Output: Advanced state vectors

PWVCON

Purpose: To perform various vector transformations needed by LSAEG

Called by: LSAEG

Comments: See reference 9

QRDTPI

Purpose: To compute the desired phase angle at TPI which corresponds to the desired elevation angle E

Called by: SOR

Input: E, ΔH_{CDH}, $X_{E,J}$

Output: θ_i, $\bar{R}'_J$, $\bar{V}'_J$

RANDN

Purpose: UNIVAC MATH-PACK routine which generates an array of random numbers that has Gaussian distribution and has a mean and standard deviation approximating a desired input

Called by: RANNO

Input: n, x_1, σ, μ

Output: $\{X_i\}_{i=1}^{n}$

Comments: RANDN accepts an input number X_1 and outputs an array $\{X_i\}_{i=1}^{n}$ which approximates a normal distribution with mean μ and standard deviation σ. The method of generation is a recursive congruence scheme; that is

$$X_{i+1} = f(X_i)\text{MOD } 2^{27}$$

On each call to RANDN, the first number generated replaces the input X_i. A detailed explanation of the generating function may be found in references 22 and 23.

RANDU

Purpose: UNIVAC MATH-PACK routine that generates an array of random numbers uniformly distributed on the interval [o,1]

Called by: RANNO, TRIM

Input: n, y_1

Output: $\{y_i\}_{i=1}^{n}$

Comments: See reference 23 for a description of the equations used in RANDU

RANNO

Purpose: To produce an array of pseudorandom numbers

Called by: AMAIN, ERRSXT, OBS, SAMPLE, SETPLT, SETRBS, SETTHR, TRIM

Input: n, I_{RAN}

Output: $\{X_i\}_{i=1}^{n}$

Comments: The UNIVAC MATH-PACK routines RANDN and RANDU are used to generate the random numbers used by the components of the AGAST program. On the first call to RANNO, the UNIVAC 1108 computer clock routine is called upon to obtain the time, which is then operated on to produce the first input to RANDN. That routine accepts X_i as the initializing parameter of the random number generation scheme. To decrease the correlation of the initializing parameter on successive calls to RANNO, the current call to that routine operates on the current output array $\{X_i\}_{i=1}^{n}$ to produce X_{RAN}, which will be the next input to RANDN.

RARP

Purpose: To compute the radius of apogee and radius of perigee of an orbit

Called by: MANEUV, OUTPUT

Comments: See reference 24

REGFAL (Regula-Falsi Iteration Routine)

Purpose: To obtain a guess at the value of the independent variable in a function that will drive the dependent variable to zero

Called by: PCA, TCONE

Input: χ, $f(x)$, I_{COVNT}, I_{GUESS}

Output: $\Delta\chi$

Comments: By use of the guess at the independent variable to recompute the dependent variable, REGFAL is recalled for a new guess. This process is continued until convergence is obtained.

REPLAC

Purpose: To compute the state vectors which start each cycle through the program and perform vector replacement events

Called by: AMAIN

Input: Initial state vector perturbation data (block B, ref. 5) and vector replacement data (blocks EVENT and LEC3, ref. 5)

Output: The desired state vectors

RNR

Purpose: To accept simulated sextant and VHF observational data and to perform navigational updates to the CMC estimated state vectors

Called by: UPDATV

Input: $X_{E,I}$, $X_{E,J}$, $\bar{W}_D$, [W], [T], [NBSM], ΔR_M, $\bar{U}_M$, VAR_{VHF}, VAR_{VHFmin}, VAR_{SXT}, VAR_{IMU}, VAR_{INT}, LM_{UP}, I_{PRINT}, KRNDFG

Output: $X_{E,I}$, [W]

Comments: Subroutine RNR is an engineering simulation of the VHF and optical navigational logic described in section 5.2.5.2 of reference 2.
The W-matrix in the RNR flow chart of appendix B is partitioned as follows.

$$[W] = \left(\begin{array}{c|c|c} W_1 & W_2 & W_3 \\ \hline W_4 & W_5 & W_6 \\ \hline W_7 & W_8 & W_9 \end{array}\right)$$

RNRPRT

Purpose: To print rendezvous tracking information: actual state vectors, estimated state vectors, tracking observations, filter parameters, and state vector corrections.

Called by: AGSRR, FILTER, ONR, RNR, RR, UPDATV

Input: J_F, I_P, N, I_{COM}

Output: Prints as indicated in the flow chart of RNRPRT in appendix B.

ROTXYZ

Purpose: To compute a rotation matrix [M] for the angles θ_x, θ_y and θ_z; the rotation is in the x-y-z order

Called by: CPEM, ERRSXT, OBS

Input: θ_x, θ_y, and θ_z

Output: [M]

Comments: The rotation matrix $[M] = (M_{ij})$ is computed as follows.

$$M_{11} = \mathrm{COS}\ \theta_z\ \mathrm{COS}\ \theta_y$$
$$M_{12} = \sin\theta_z \cos\theta_x + \cos\theta_z \sin\theta_y \sin\theta_x$$
$$M_{13} = \sin\theta_z \sin\theta_x - \cos\theta_z \sin\theta_y \cos\theta_x$$
$$M_{21} = - \sin\theta_z \cos\theta_y$$
$$M_{22} = \cos\theta_z \cos\theta_x - \sin\theta_z \sin\theta_y \sin\theta_x$$
$$M_{23} = \cos\theta_z \sin\theta_x + \sin\theta_z \sin\theta_y \cos\theta_x$$
$$M_{31} = \sin\theta_y$$
$$M_{32} = - \cos\theta_y \sin\theta_x$$
$$M_{33} = \cos\theta_y \cos\theta_x$$

RR

Purpose: To accept simulated rendezvous radar observational data and to perform navigational updates to LGC estimated state vectors

Called by: UPDATV

Input: $X_{E,I}$, $X_{E,J}$, $\bar{W}_D$, [W], [T], [SMNB], ΔR_M, $\Delta\dot{R}_M$, β_M, θ_M, VAR_R, VAR_{RMIN}, VAR_V, VAR_{VMIN}, VAR_β, VAR_{IMU}, $\delta\beta_E$, $\delta\theta_E$, LM_{UP}, I_{PRINT}, I_{ANGLE}, KRNDFG

Output: $\chi_{E,I}$, $[W]$, $\delta\beta_E$, $\delta\theta_E$

Comments: Subroutine RR is an engineering simulation of the logic described in section 5.2.4.2.2 of reference 3.
The W-matrix of the RR flow chart of appendix B is partitioned as follows.

$$[W] = \left(\begin{array}{c|c|c} W_1 & W_2 & W_3 \\ \hline W_4 & W_5 & W_6 \\ \hline W_7 & W_8 & W_9 \end{array}\right)$$

RRSMNB

Purpose: To compute the stable member to navigation base rotation matrix for the LM

Called by: AGSRR, UPDATV

Input: $\Delta\bar{R}$, $\bar{P}$, $[T_A]$

Output: [SMNB]

Comments: The matrix $[T_A]$ is a transformation from basic reference coordinates to the true platform orientation. The output matrix [SMNB] represents an orientation of the LM with the z-body axis directed along the relative line of sight to the CSM, the y-body axis directed perpendicular to the plane defined by the LM position vector and the line of sight vector, and with the x-body axis completing a right-handed system.

RTCCSI

Purpose: To compute the concentric sequence initiation maneuver based on the RTCC logic, and to compute the NC1 - NH sequence in the DKI maneuver sequences

Called by: COMDKI, TARGET

Input: T_{CSI}, T_{CDH}, T_{TPI}, E_L, N_{CDH}, I_{VEH}, L_{EN}, Δh_D

Output: $\overline{\Delta V}_{CSI}$ (LV), T_{CDH}, $\overline{\Delta V}_{CDH}$ (LV)

Comments: The logic of RTCCSI is a combination of the CSI/CDH logic of references 2 and 25

RUNGA

Purpose: To integrate the actual state vector through each guidance cycle of a burn and to compute the thrust model used during each guidance cycle of the burn

Called by: MANEUV

Input: $\bar{R}$, $\bar{V}$, T_{GO}, $\bar{U}_{TD}$, T, I, I_{ENG}, I_{PH}, δt, E_{NO}, W, L_{EN}, K_{TOFF}, I_{MDC}, V_{FLAG}, N_{THROT}, K_{UL}, F_D, I_{SP_D}, $\Delta\bar{V}_{TOTAL}$, C_K, θ_{CANT}, F^*_A , $I^*_{SP_A}$, I_{ME}, I^*_{THRC}, ΔT^*_{BP}

Output: $\bar{R}$, $\bar{V}$, $\Delta\bar{V}$, $\bar{U}_{TA}$, T, ΔT_{BURN}, I_{PH}, W, K_{TOFF}, K_{UL}, $\dot{W}$, ΔV_{TOTAL}, C_K

SAMCOV

Purpose: To accept an array of random dispersions to a state vector and to compute a sample covariance matrix, correlation coefficients, standard deviations of the individual components of the state vector, and standard deviations for the position and velocity

Called by: PROCES

Input: δX_S

Output: Σ, $\{\sigma_i\}_{i=1}^{6}$, σ_R, σ_V

Comments: The output Σ has in its lower left triangular part the correlation coefficients

SAMDIF

Purpose: To compute for a reference point the differences between sample arrays of state vectors produced for that reference point. The entries in the table of state vector differences are output in a local vertical coordinate system

Called by: PROCES

Input: N, I_{DIF}, X_{R,I_V}, $X_{A,1}$, $X_{A,2}$, $X_{E,1}$, $X_{E,2}$

Output: $\{\delta X_k\}_{k=1}^{n}$

SAMPLE

Purpose: To sample a covariance matrix to produce a sample random dispersion of the vector for which the covariance matrix is defined

Called by: AFLPN, REPLAC

Input: K_D, Σ, X_R

Output: X, δX

Comments: See section 7.1 of this document for a detailed explanation of the sampling technique used in SAMPLE

SAMSIG

Purpose: To compute sample means and standard deviations for an array of random variables

Called by: PROCES

Input: K, N, I_{SIG}, $\{\{X_{ji}\}_{j=1}^{n}\}_{i=1}^{k}$

Output: $\{\delta\mu_i\}_{i=1}^{k}$, $\{\sigma_i\}_{i=1}^{k}$

Comments: The formulation used in SAMSIG is based on derivations in reference 7

SAVIP2

Purpose: To read the saved data from unit 15 back into the program needed to initialize each cycle and to perform each event

Called by: AMAIN

Input: L_{EC}

Output: The saved INPUT1 data or the saved event package input

SCTXYZ

Purpose: To convert a position vector in spherical coordinates to Cartesian coordinates

Called by: INPUT1

Input: R, λ, ϕ

Output: x, y, z

Comments: The conversion is computed as follows

$$x = R \cos \lambda \cos \phi$$

$$y = R \sin \lambda \cos \phi$$

$$z = R \sin \phi$$

SENSOR

Purpose: To obtain the sensed ΔV during each guidance cycle of a maneuver

Called by: RUNGA

Input: $\bar{R}$, $\bar{U}_{TD}$, $\bar{\omega}_C$, W, $\dot{W}$, F_A

Output: $\bar{\Delta V}$, $\bar{U}_{TA}$

Comments: See reference 26

SETPC

Purpose: To load the constant array with the proper data depending upon which central body is being considered

Called by: INPUT

Input: K_{REF}

Output: π, R_{REF}, μ, J_2, J_2, J_3, J_4, J_{22}, J_{31}

SETPLT

Purpose: To set the initial platform misalinement angles, gyro drift rates, accelerometer misalinement angles, accelerometer read biases, and accelerometer scale factors

Called by: ALIGN, AMAIN

Input: I_{PE}, I_{PLTSG}, I_{PLTN}, σ_{ALN}, σ_{DFT}, σ_{AAL}, σ_{DVB}, σ_S

Output: $\theta_{x,o}$, $\theta_{y,o}$, $\theta_{z,o}$, $\dot{\theta}_x$, $\dot{\theta}_y$, $\dot{\theta}_{z,}$ ε-array, $\Delta\bar{V}_B$, S-array

SETRBS

Purpose: To set RR and SXT bias values

Called by: AMAIN

Input: $\sigma_{\delta R}$, $\sigma_{\delta \dot{R}}$, $\sigma_{\delta \beta}$, $\sigma_{\delta \theta}$, σB_{SXT_x}, σB_{SXT_y}, σB_{SXT_z}, σB_{VHF}

Output: δR, $\delta \dot{R}$, $\delta \beta$, $\delta \theta$, δ_{SXT_x}, δ_{SXT_y}, δ_{SXT_z}, VHF_{BIAS}, RBIAS

SETTHR

Purpose: To set for each thrust phase of each engine the thrust level and specific impulse which model the actual engine performance

Called by: AMAIN

Input: J_1, J_2, K_1, K_2, I_{NOMT}, I_{THR}, F_A*, I_{SPA}*

Output: F_A, I_{SPA}

SHORTB

Purpose: To compute an estimated burn time to determine if steering and in the case of the DPS engine if throttling is allowed

Called by: MANEUV

Input: I_{VEH}, I_{ENG}, $|\bar{V}_G|$, ENO, W_E, $\dot{W}_E$

Output: K_{STBURN}, N_{THROT}, T_{GO}

Comments: See the time-to-go predictor routine in references 2 and 3

SMEAN

Purpose: To accept an array of state vectors to compute the mean vector of the array, and to compute the magnitude of the mean position vector and the mean velocity vector

Called by: PROCES

Input: N, $\{\delta X_k\}_{k=1}^{n}$

Output: $\bar{\mu}$, μ_R, μ_V

SOR

Purpose: This routine contains the Stable Orbit Rendezvous logic of section 5.4.2.6 of reference 3: targeting for the SOI, SOR, and SOM maneuvers. Also contained in SOR is logic to compute for the active vehicle desired offsets from the passive vehicle at CDH time, and to subsequently compute the SOI and SOR maneuvers to achieve those offsets at CDH and to maneuver into a coelliptic orbit

Called by: TARGET

Input: L_{EN}, K_{TARG}, δr, ΔH_D, ωt, T, $X_{E,I}$, $X_{E,J}$, T_{TPI}, ε_D, ε_A

Output: $\Delta\bar{V}_1$, $\Delta\bar{V}_2$, T_1, T_2

SPHELE

Purpose: To compute spherical orbit elements from Cartesian elements and vice versa

Called by: AEG, ENCKE, LSAEG

Comments: See subroutine CONV of reference 17

SPLIT

Purpose: To call subroutine START to compute the burn time for an external ΔV maneuver, then to compute the time of ignition and bias the external $\Delta\bar{V}$ by one half the finite burn arc

Called by: MANEUV

Input: I_{ENG}, T_{EX}, $\Delta\bar{V}_{EX}$, $\chi_{E,I}$, E_{NO}, F^*_A, $I^*_{SP_A}$, $\Delta T^*_{BP_A}$

Output: $\Delta\bar{V}_{EX}$

START

Purpose: To calculate the thrust ignition time, the burn duration, and the thrust attitude required of the actual maneuver which duplicates the results obtained by a given impulsive maneuver

Called by: SPLIT

Input: ΔV, W_T, ΔT_{TO}, F, $\dot{W}$, ΔT_{BP}, I_{ENG}

Output: ΔT_{TI}, ΔT_B, P, Y

STAYT

Purpose: To propagate a vector through the time interval T_{21} by use of Keplerian motion

Called by: CONIC, TIMEP

Input: $\bar{R}(T_1)$, $\bar{V}(T_1)$, χ, ξ, $C(\xi)$, $S(\xi)$, T_{21}

Output: $\bar{R}\ (T_2)$, $\bar{V}\ (T_2)$

Comments: See reference 2, pages 5.5-42 and 5.5-48

TA

Purpose: To compute true anomaly when eccentricity and mean anomaly are given

Called: AEG, ENCKE, ETOXYZ, MANEUV, OUTPUT, RARP

Input: e, M

Output: θ

TAPEUP

Purpose: To initialize the JPL ephemeris tape

Called by: INPUT

Input: Base date

TARGET

Purpose: To target all desired maneuvers

Called by: MANEUV

Input: Block EVENT and Block LECI data (see ref. 5)

Output: T_{IG}, $\overline{\Delta V}_{EX}$, I_{FLAG}, I_{SPLIT}

Comments: The routine has the capability to target the maneuver on the primary computer states and to target each maneuver based on the actual states and on the estimated states of other computers. All the solutions requested for a given maneuver are stored in the SAVMAN array to be used as described in section 6.5.2 of this document.

TAUAL

Purpose: To compute the time lag and ΔH between two orbiting vehicles

Called by: PMISS

Comments: See reference 27 for a discussion of the formulation of subroutine TAUAL

TCONE

Purpose: To compute the time of closest approach of the CSM to the landing site

Called by: AFLPN

Input: T, vehicle number

Output: T_1

THETR

Purpose: To compute the phase angle between two orbiting vehicles

Called by: CFAEG5, JOCDH, MANEUV, PMISS, RTCCSI, TARGET, TAUAL

Input: Orbital elements i, h, and u for both vehicles

Output: θ_R

Comments: For an explanation of the formulation of THETR, see reference 27

THRFIL

Purpose: To accept the simulated accelerometer readings and to output the acceleration magnitude and τ

Called by: MANEUV

Input: ΔT, ΔV, τ, V_E

Output: a_m, τ

TILL

Purpose: To compute the time of intersection of a spacecraft's trajectory with a right circular cone whose vertex is centered in the planet

Called by: TCONE

Comments: See reference 28

TIMAL

Purpose: To determine the arrival time of a spacecraft at a designated counter line

Called by: COMDKI, PMISS, RTCCSI

Comments: A discussion of the formulation of TIMAL occurs in reference 27

TIMEB

Purpose: To compute the burn time for a maneuver

Called by: START

Input: ΔV, W_o, $\dot{W}$, g, I_{SP}

Output: ΔT_B

Comments: The burn time is commputed as follows.

$$\Delta T_B = \frac{W_o}{\dot{W}} \left[1 - \frac{1}{\text{EXP}\left(\frac{|\Delta V|}{g I_{SP}}\right)} \right]$$

TIMEP (Time-Theta Routine)

Purpose: To compute the time required for a vehicle to pass through an angle θ using Keplerian motion

Called by: CFP, LOPC, PRETPI, QRDTPI, SOR

Input: $\bar{R}\ (T_1)$, $\bar{V}\ (T_1)$, θ

Output: T_{21}

Comments: See the Time-Theta Routine of reference 2, page 5.5-34

TIMFAL

Purpose: To compute the time of arrival of a spacecraft at an upcoming apsis

Called by: COMDKI, RTCCSI

Comment: This routine is documented in reference 27

TPASS

Purpose: To compute a table of times for which an orbiting vehicle passes within sight of a lunar landmark

Called by: CATIME

Input: T, T_{START}, T_{STOP}, $\phi_{R,O}$, $\lambda_{R,O}$, ϕ_R, λ_R, H, R_{REF}, $\bar{R}_{LSA}$, $X_{A,I}$, ε_{MIN}, ΔT_{PASS}

Output: $\{T_{SAVE}\ (i)\}_{i=1}^{N}$, $\{\Delta T_{SAVE}\ (i)\}_{i=1}^{N}$

Comments: The iterative scheme used in TPASS to compute the closest approach times is documented in reference 29

TPITPM

Purpose: To display the maneuver ΔV's associated with maneuvers that use the onboard Lambert targeting

Called by: AGS, MIDCRS, PRETPI, SOR

Input: T_1, T_2, $\bar{\Delta V}_1$, $\bar{\Delta V}_2$, R_I, V_I, R_J, V_J, I_{PC}

Output: The Lambert targeting display

TRANCO

Purpose: To perform transformations of vectors from planetary coordinates to inertial coordinates and vice versa

Called by: AFLPN, ALIGN, DESCNT, DOI, ENCKE, EPERT, LOPC, MANEUV, ONR, TPASS, UPDATV

Input: T, $\bar{X}$, $\bar{V}$, K_{REF}, option flag

Output: $\bar{X}_R$, $\bar{V}_R$

TRIM

Purpose: To compute and trim the V_G residuals caused by initial thrust misalinement and center of gravity offset errors

Called by: MANEUV

Input: X_A, X_E, X_M, $\bar{U}_{TA}$, $\bar{V}_G$, I_{EX}

Output: X_A, X_E, X_M, $\bar{V}_G$

TURNUP

Purpose: To compute an optimum ΔV two-impulse solution based on the interpolation technique of subroutine GUESS

Called by: BUTTON

Comments: No documentation is available

ULLDIR

Purpose: Based on its knowledge of the center of gravity location this subroutine calculates the thrust direction during ullage and the initial thrust direction of the main engine after ullage

Called by: MANEUV, RUNGA

Input: $\bar{U}_{TD}$, $\bar{U}_{TA}$, X_E, W_T, I_{VEH}

Output: $\bar{U}_{TD}$, $\bar{U}_{TA}$

UNIT

Purpose: To compute the unit vector $\bar{U}$ which corresponds to an input vector $\bar{a}$

UNIVAR

Purpose: To compute the universal parameter x used by subroutine LAMBRT and TIMEP

Called by: LAMBRT, TIMEP

Comments: See section 5.5.10 of reference 2

UPDATV

Purpose: Driver logic for the onboard tracking programs

Called by: AMAIN

Input: I_{MARK}, L_{EN}, ΔT_{MARK}, T_{START}, T_{STOP}, IVUP, [T], I_{PE}, I_{GIM}, ε_T, landing site array, $X_{E,1}$ $X_{E,2}$, $X_{A,1}$, $X_{A,2}$

Output: $X_{E,IVUP}$

VECMG

Purpose: To compute the magnitude of an input vector $\bar{a}$

VTBG (Velocity-To-Be-Gained Routine)

Purpose: To obtain $\bar{V}_G$ for each guidance cycle of a Lambert guided burn

Called by: MANEUV

Input: $\bar{R}$, $\bar{V}$, N, N_{MAX}, $\bar{g}$, $\bar{g}_b$, $\bar{\Delta}\bar{V}_E$, T_{IG}, $\bar{b}\bar{\Delta}T$, T_2, $\bar{R}_T$

Output: $\bar{V}_G$

Comments: See reference 3, section 5.3.3.5

WMATCN

Purpose: To transform the [W] used by ONR into a triangular matrix; this form is required to propagate [W] to the next landmark

Called by: ONR

Input: [W]

Output: [W]

WMT

Purpose: To print the W-matrix, the E-matrix, and the square roots of the diagonal of the E-matrix

Called by: RNRPRT

Input: $\bar{W}_C$, [BRLOS], I_{PRINT}

Output: None

Comments: The W-matrix is input to WMT as a column vector with 81 entries, $\bar{W}_C$. That vector is loaded_into the W-matrix array [W]. The E-matrix is computed as WW^T in LOS cordinates.

XESTAB

Purpose: To maintain a table of current state vectors for each vehicle of each computer

Called by: AFLPN, AGSRR, ALIGN, AMAIN, COMDKI, DELTAV, MANEUV, PROPAG, REPLAC, TARGET, UPDATV

Input: X_E, I_{COM}, I_{VEH}

Output: X_E

XMAIN

Purpose: A driver program which calls subroutine PROCES to process a data tape generated by AGAST on a previous computer run

Input: N_{SAM}, N_{REF}

Output: Same as subroutine PROCES

XYZTOE

Purpose: To convert Cartesian orbit elements to Keplerian orbit elements

Called by: AEG, AGSKEP, CFP, ENCKE, LSAEG, MANEUV

Input: $\bar{R}$, $\bar{V}$

Output: a, e, i, g, h, M

XYZTSC

Purpose: To convert a position vector to spherical coordinates

Called by: AFLPN

Input: x, y, z

Output: R, λ, ϕ

Comments: The conversion is computed as follows.

$$R = \sqrt{x^2 + y^2 + z^2}$$

$$\lambda = \tan^{-1}\left(\frac{y}{x}\right)$$

$$\phi = \cos^{-1}\left(\frac{Z}{\sqrt{x^2 + y^2}}\right)$$

APPENDIX B

FLOW CHARTS OF THE SUBROUTINE OF AGAST

APPENDIX B - FLOW CHARTS OF THE SUBROUTINE OF AGAST

The following figures are in alphabetical order according to the subroutine names. The driver program is the routine AMAIN. Not all routines used in AGAST have flow charts in this appendix; however, documentation of the larger routines not flow-charted here occurs in the references noted in the subroutine descriptions of appendix A. Symbols used in these flow charts are defined in appendix C.

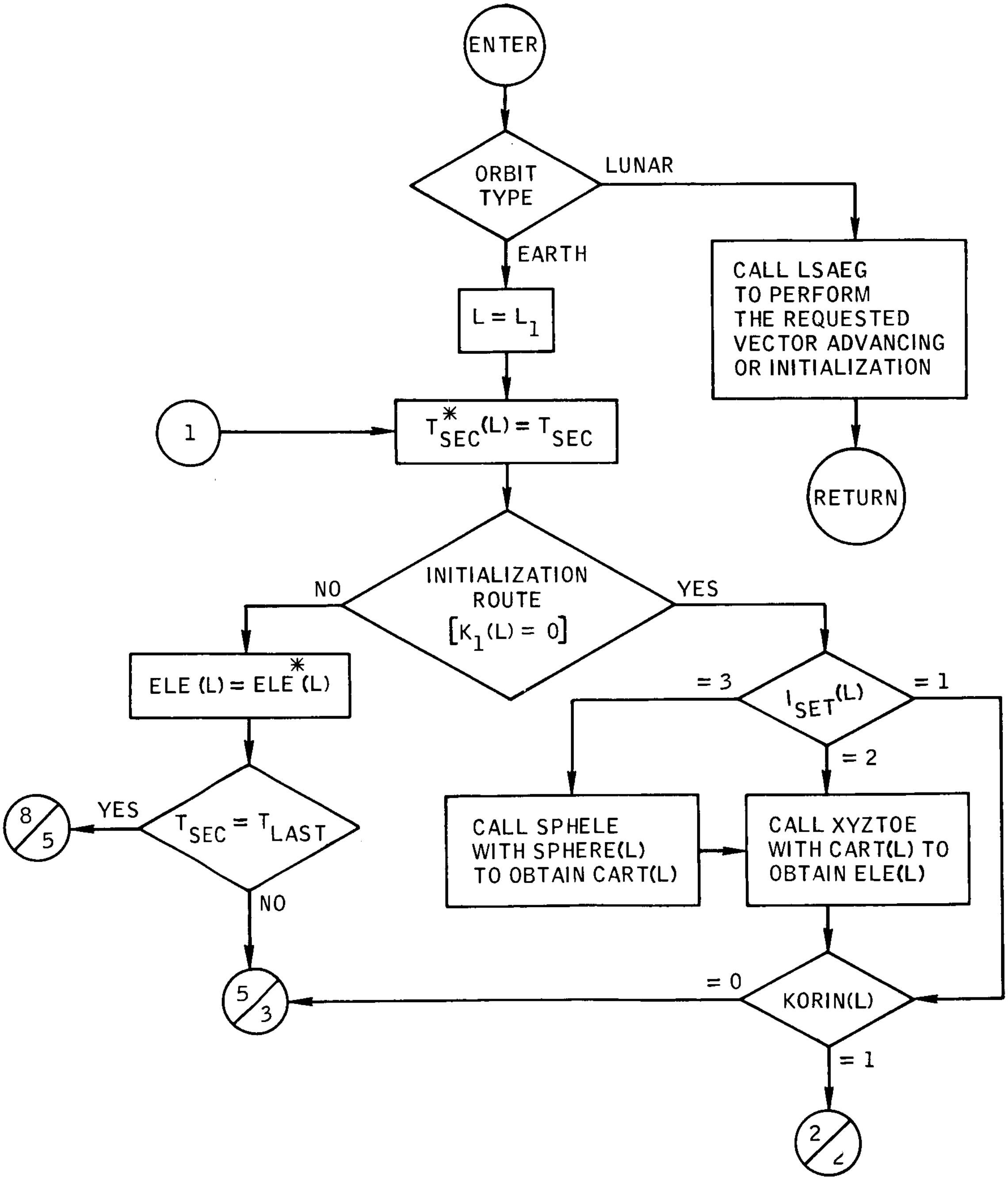
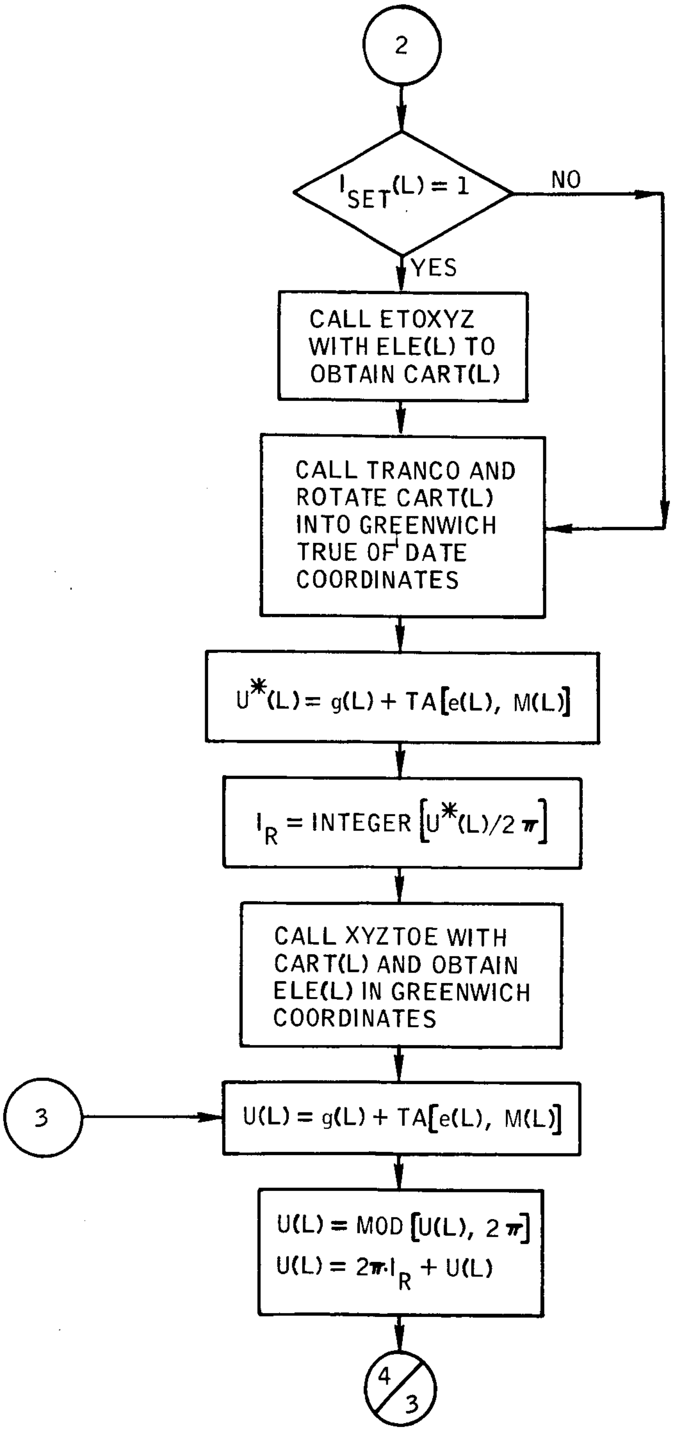

ENTER
ORBIT TYPE
LUNAR
EARTH
CALL LSAEG TO PERFORM THE REQUESTED VECTOR ADVANCING OR INITIALIZATION
RETURN
$L = L_1$
1
$T^*_{SEC}(L) = T_{SEC}$
INITIALIZATION ROUTE $[K_1(L) = 0]$
NO
YES
$ELE(L) = ELE^*(L)$
$I_{SET}(L)$
= 3
= 1
= 2
CALL SPHELE WITH SPHERE(L) TO OBTAIN CART(L)
CALL XYZTOE WITH CART(L) TO OBTAIN ELE(L)
$T_{SEC} = T_{LAST}$
YES
NO
8/5
5/3
KORIN(L)
= 0
= 1
2/2

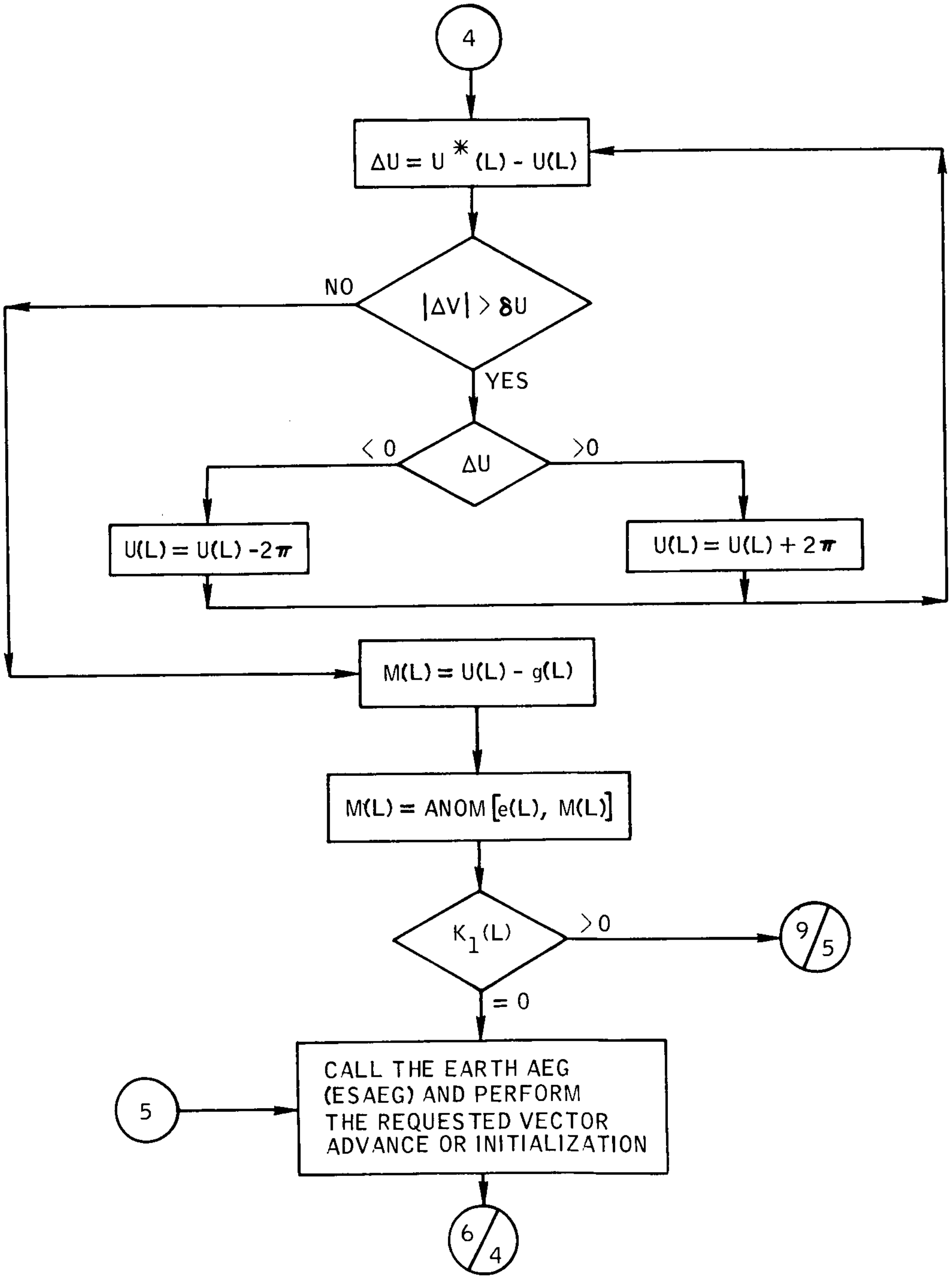
Figure B-1.- Flow chart of subroutine AEG.

Page 1 of 5


2
$I_{SET}(L) = 1$
NO
YES
CALL ETOXYZ WITH ELE(L) TO OBTAIN CART(L)
CALL TRANCO AND ROTATE CART(L) INTO GREENWICH TRUE OF DATE COORDINATES
$U^*(L) = g(L) + TA[e(L), M(L)]$
$I_R = \text{INTEGER}[U^*(L)/2\pi]$
CALL XYZTOE WITH CART(L) AND OBTAIN ELE(L) IN GREENWICH COORDINATES
3
$U(L) = g(L) + TA[e(L), M(L)]$
$U(L) = \text{MOD}[U(L), 2\pi]$
$U(L) = 2\pi \cdot I_R + U(L)$
4
3

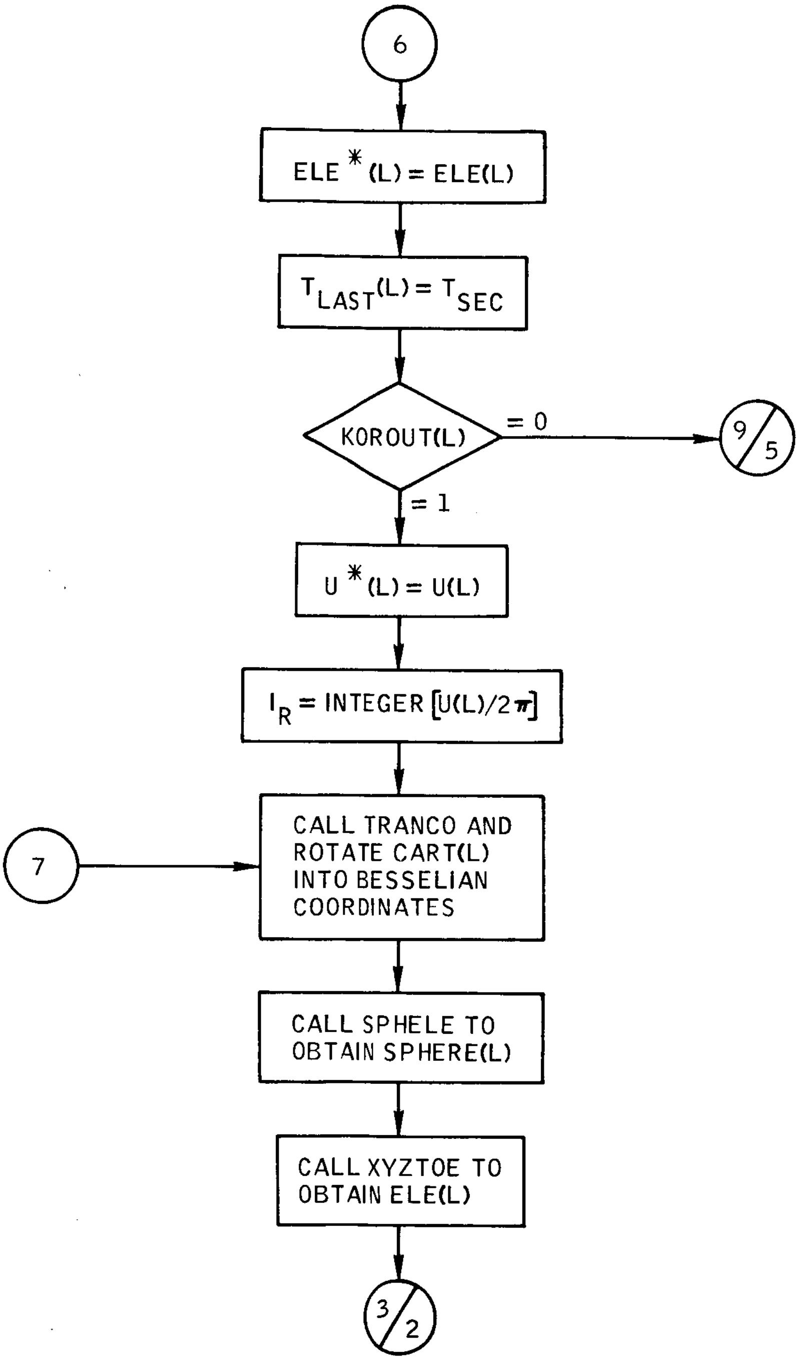
Page 2 of 5

Figure B-1.- Continued.



Page 3 of 5

Figure B-1.- Continued.


6
ELE * (L) = ELE(L)
T LAST (L) = T SEC
KOROUT(L)
= 0
9/5
= 1
U * (L) = U(L)
I R = INTEGER [U(L)/2π]
7
CALL TRANCO AND ROTATE CART(L) INTO BESSELIAN COORDINATES
CALL SPHELE TO OBTAIN SPHERE(L)
CALL XYZTOE TO OBTAIN ELE(L)
3/2

Page 4 of 5

Figure B-1.- Continued.

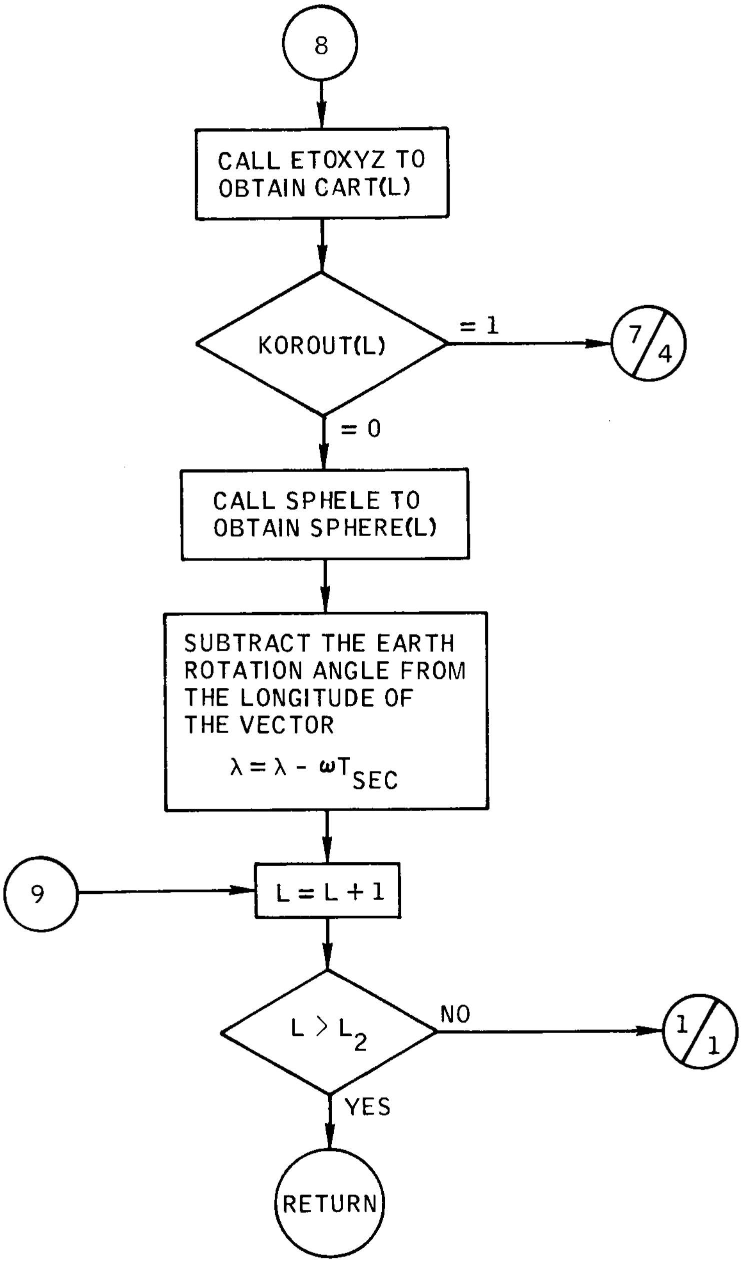
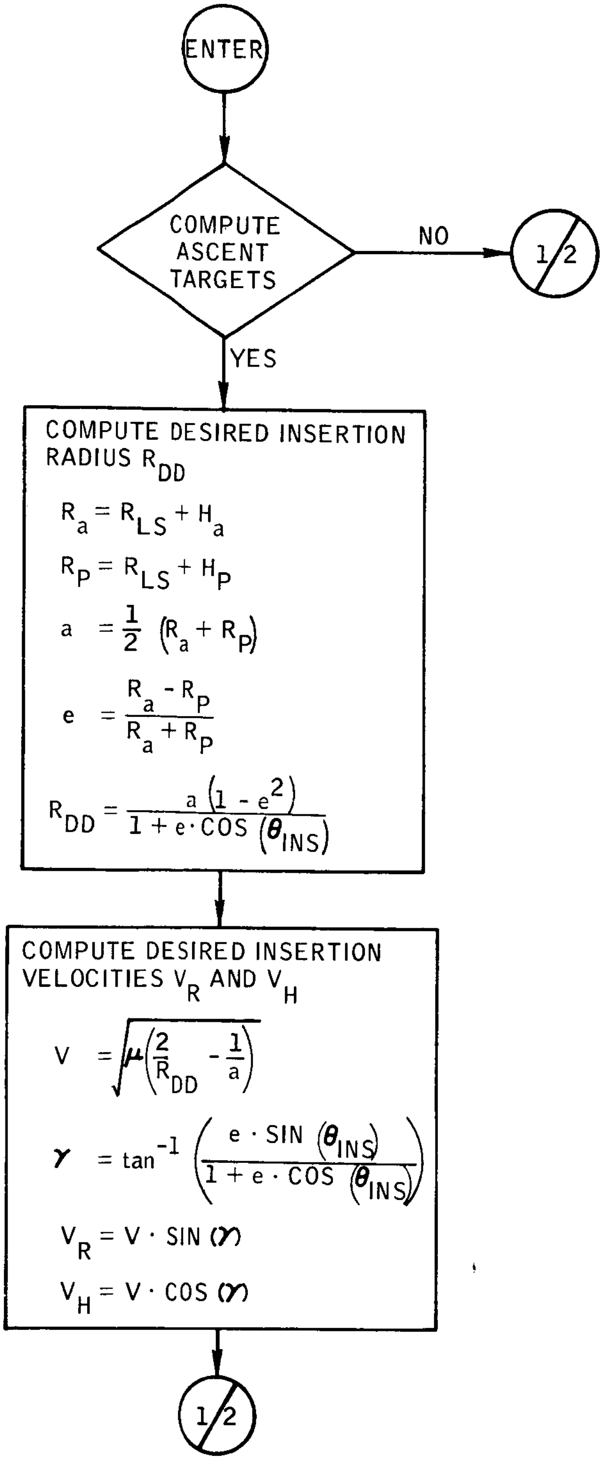

8
CALL ETOXYZ TO OBTAIN CART(L)
KOROUT(L)
= 1
7/4
= 0
CALL SPHELE TO OBTAIN SPHERE(L)
SUBTRACT THE EARTH ROTATION ANGLE FROM THE LONGITUDE OF THE VECTOR
$\lambda = \lambda - \omega T_{SEC}$
9
L = L + 1
L > L_2
NO
1/1
YES
RETURN

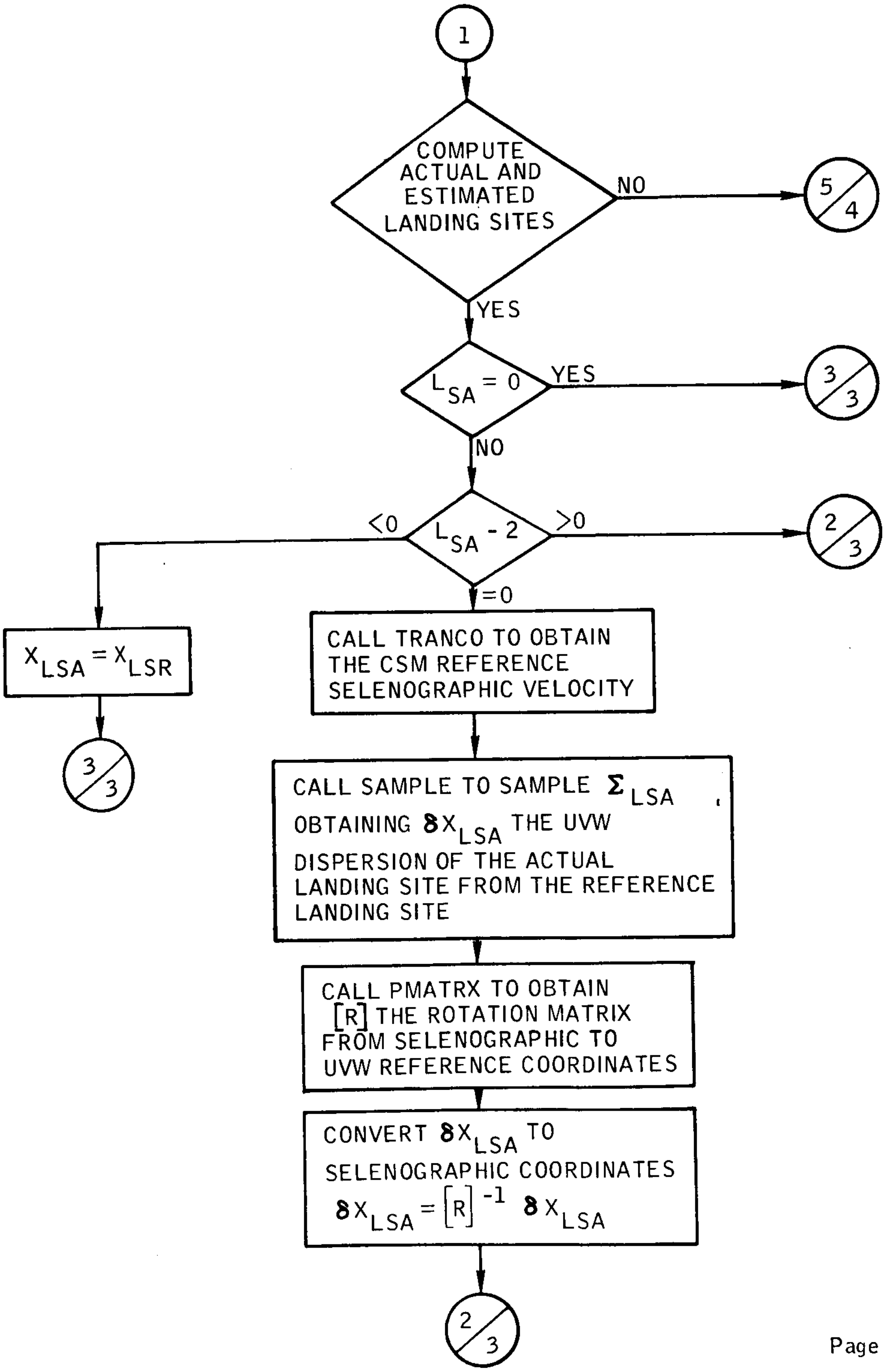
Figure B-1.- Concluded.

Page 5 of 5


ENTER
COMPUTE ASCENT TARGETS
NO
1/2
YES
COMPUTE DESIRED INSERTION RADIUS R_{DD}
$R_a = R_{LS} + H_a$
$R_P = R_{LS} + H_P$
$a = \frac{1}{2} \left(R_a + R_P\right)$
$e = \frac{R_a - R_P}{R_a + R_P}$
$R_{DD} = \frac{a\left(1 - e^2\right)}{1 + e \cdot COS\left(\theta_{INS}\right)}$
COMPUTE DESIRED INSERTION VELOCITIES V_R AND V_H
$V = \sqrt{\mu\left(\frac{2}{R_{DD}} - \frac{1}{a}\right)}$
$\gamma = \tan^{-1}\left(\frac{e \cdot SIN\left(\theta_{INS}\right)}{1 + e \cdot COS\left(\theta_{INS}\right)}\right)$
$V_R = V \cdot SIN(\gamma)$
$V_H = V \cdot COS(\gamma)$
1/2

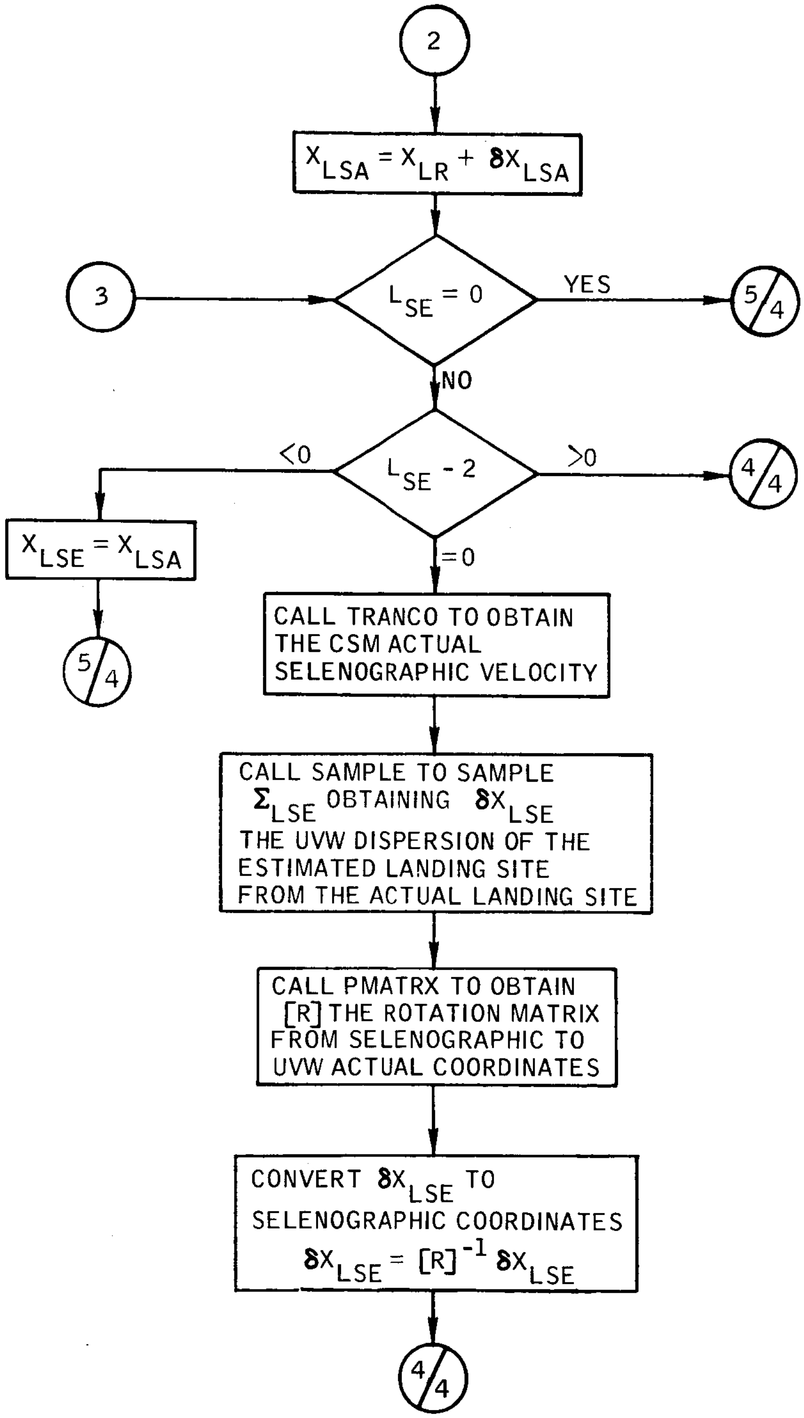
Page 1 of 6

Figure B-2.- Flow chart of subroutine AFLPN.


1
COMPUTE ACTUAL AND ESTIMATED LANDING SITES
NO
5/4
YES
$L_{SA} = 0$
YES
3/3
NO
<0
$L_{SA} - 2$
>0
2/3
=0
$X_{LSA} = X_{LSR}$
3/3
CALL TRANCO TO OBTAIN THE CSM REFERENCE SELENOGRAPHIC VELOCITY
CALL SAMPLE TO SAMPLE Σ_{LSA} OBTAINING δX_{LSA} THE UVW DISPERSION OF THE ACTUAL LANDING SITE FROM THE REFERENCE LANDING SITE
CALL PMATRX TO OBTAIN [R] THE ROTATION MATRIX FROM SELENOGRAPHIC TO UVW REFERENCE COORDINATES
CONVERT δX_{LSA} TO SELENOGRAPHIC COORDINATES $\delta X_{LSA} = [R]^{-1} \delta X_{LSA}$
2/3

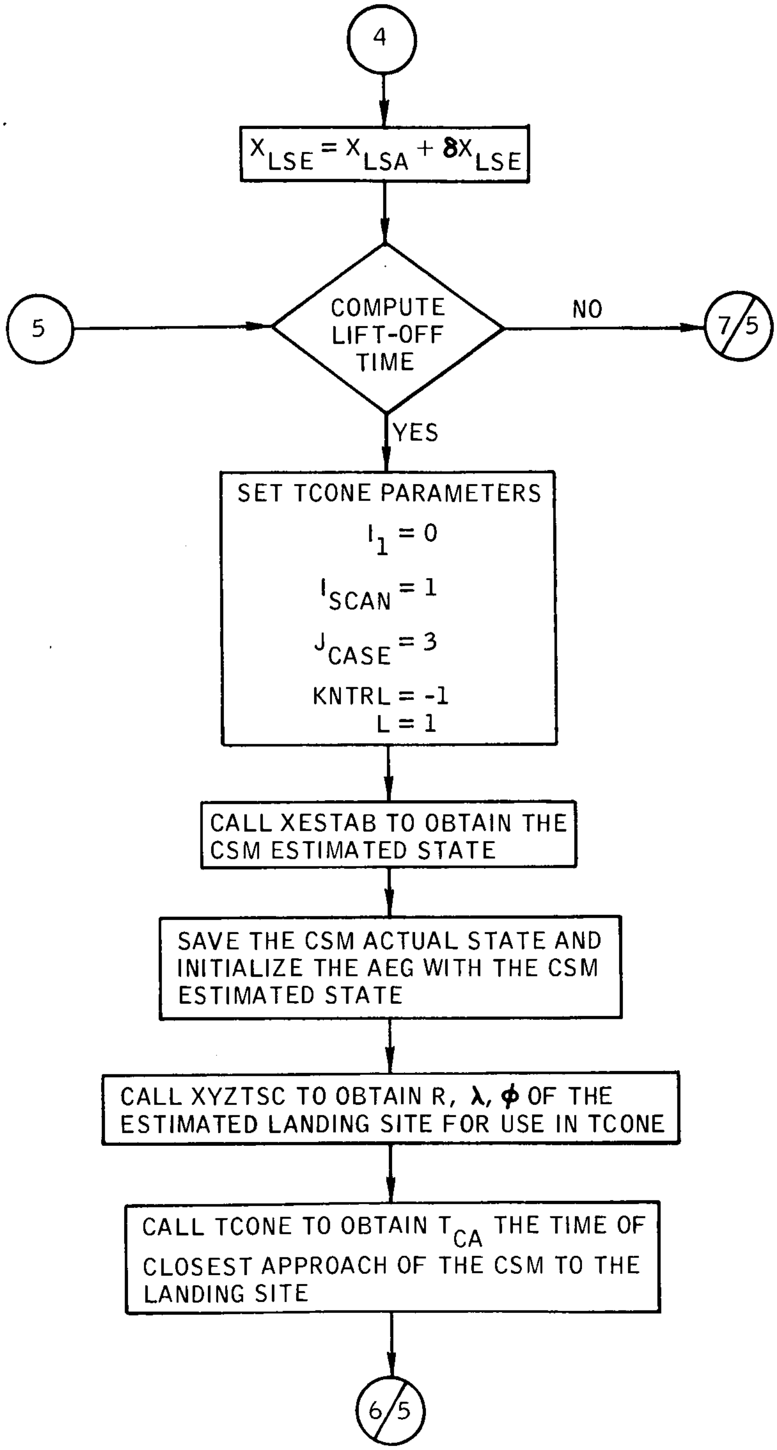
Page 2 of 6

Figure B-2.- Continued.


2
$X_{LSA} = X_{LR} + \delta X_{LSA}$
3
$L_{SE} = 0$
YES
5/4
NO
<0
$L_{SE} - 2$
>0
4/4
$X_{LSE} = X_{LSA}$
=0
5/4
CALL TRANCO TO OBTAIN THE CSM ACTUAL SELENOGRAPHIC VELOCITY
CALL SAMPLE TO SAMPLE Σ_{LSE} OBTAINING δX_{LSE} THE UVW DISPERSION OF THE ESTIMATED LANDING SITE FROM THE ACTUAL LANDING SITE
CALL PMATRX TO OBTAIN [R] THE ROTATION MATRIX FROM SELENOGRAPHIC TO UVW ACTUAL COORDINATES
CONVERT δX_{LSE} TO SELENOGRAPHIC COORDINATES $\delta X_{LSE} = [R]^{-1} \delta X_{LSE}$
4/4

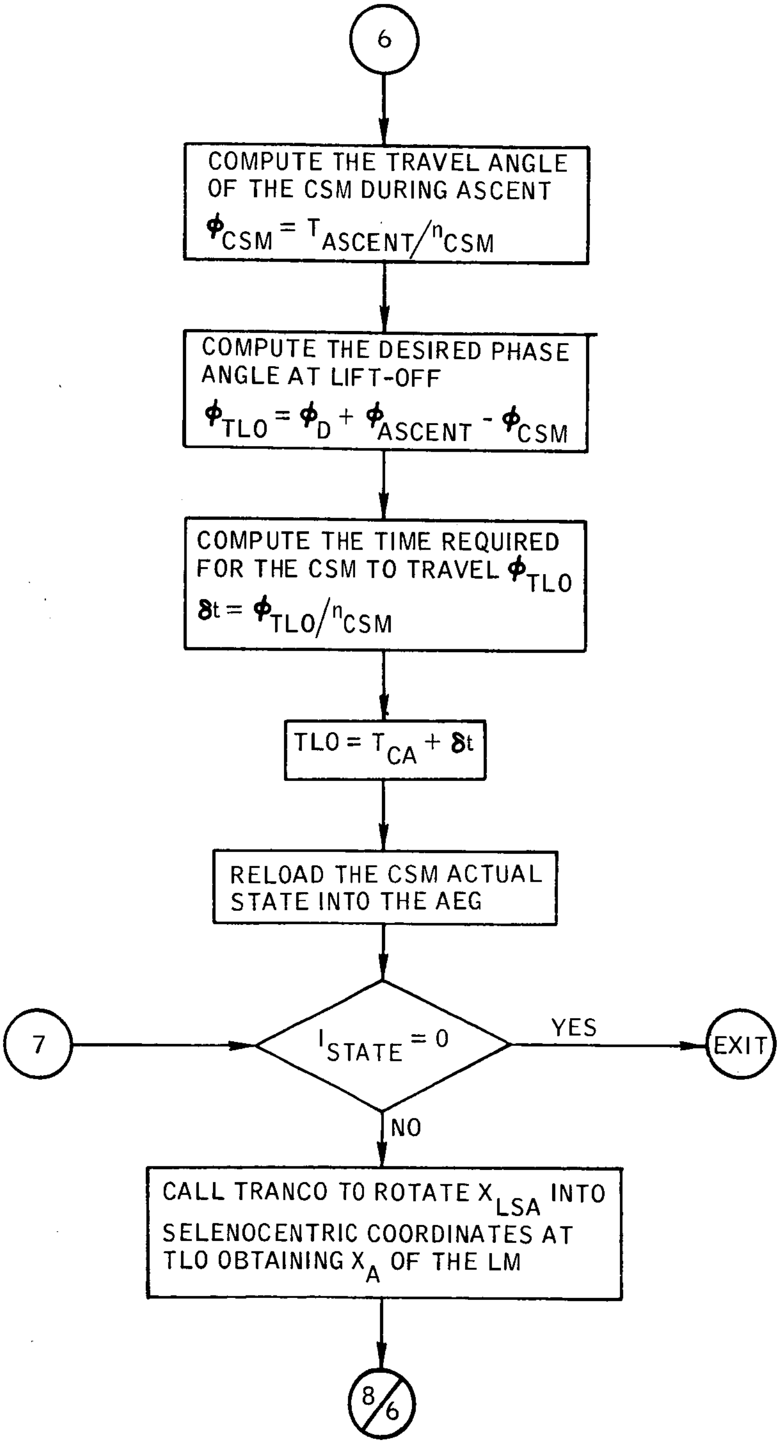
Page 3 of 6

Figure B-2.- Continued.


4
$X_{LSE} = X_{LSA} + \delta X_{LSE}$
5
COMPUTE LIFT-OFF TIME
NO
7/5
YES
SET TCONE PARAMETERS
$I_1 = 0$
$I_{SCAN} = 1$
$J_{CASE} = 3$
KNTRL = -1
L = 1
CALL XESTAB TO OBTAIN THE CSM ESTIMATED STATE
SAVE THE CSM ACTUAL STATE AND INITIALIZE THE AEG WITH THE CSM ESTIMATED STATE
CALL XYZTSC TO OBTAIN R, λ, ϕ OF THE ESTIMATED LANDING SITE FOR USE IN TCONE
CALL TCONE TO OBTAIN T_{CA} THE TIME OF CLOSEST APPROACH OF THE CSM TO THE LANDING SITE
6/5

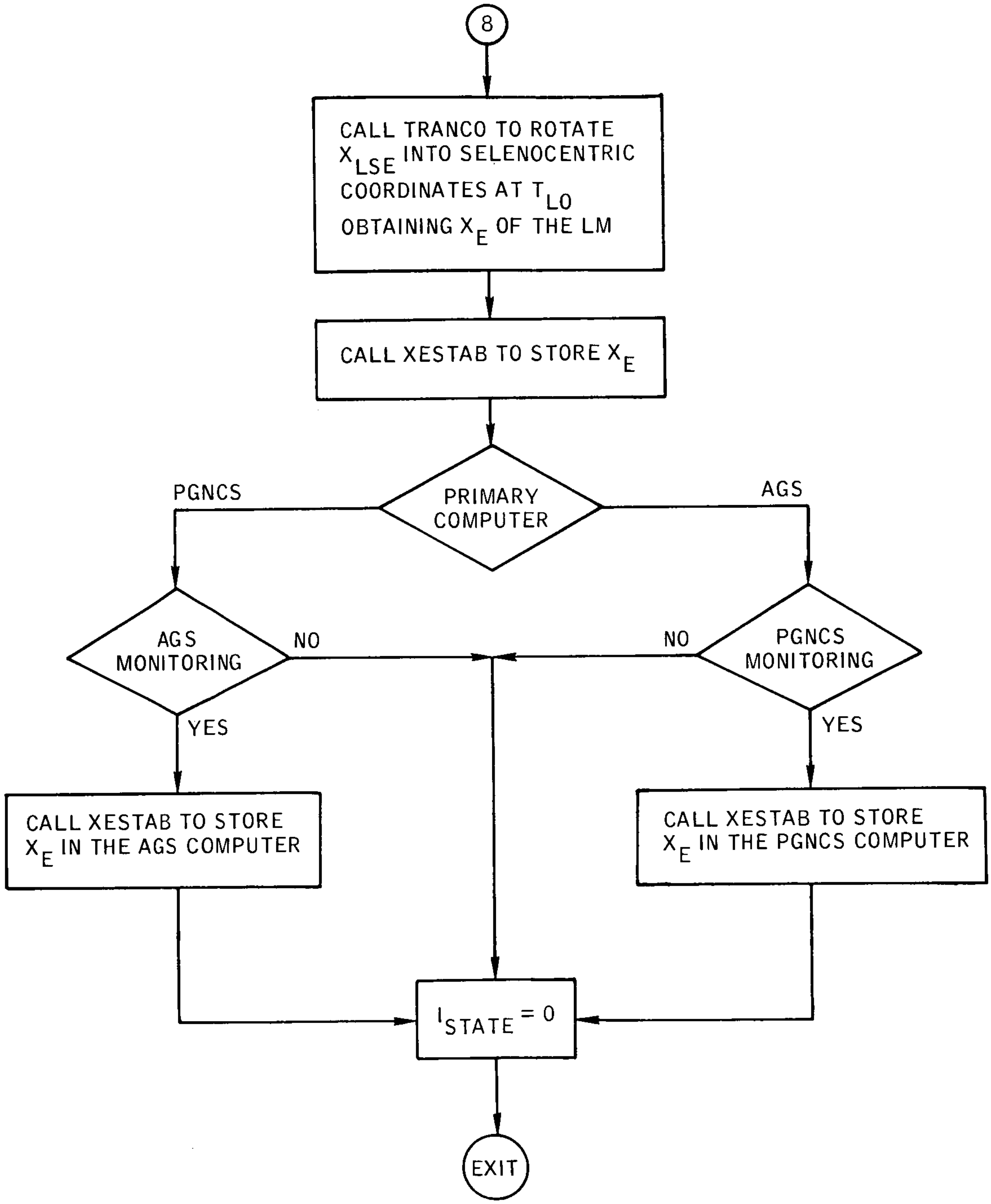
Page 4 of 6

Figure B-2.- Continued.


6
COMPUTE THE TRAVEL ANGLE OF THE CSM DURING ASCENT
$\phi_{CSM} = T_{ASCENT}/n_{CSM}$
COMPUTE THE DESIRED PHASE ANGLE AT LIFT-OFF
$\phi_{TLO} = \phi_D + \phi_{ASCENT} - \phi_{CSM}$
COMPUTE THE TIME REQUIRED FOR THE CSM TO TRAVEL ϕ_{TLO}
$\delta t = \phi_{TLO}/n_{CSM}$
$TLO = T_{CA} + \delta t$
RELOAD THE CSM ACTUAL STATE INTO THE AEG
7
$I_{STATE} = 0$
YES
EXIT
NO
CALL TRANCO TO ROTATE X_{LSA} INTO SELENOCENTRIC COORDINATES AT TLO OBTAINING X_A OF THE LM
8/6

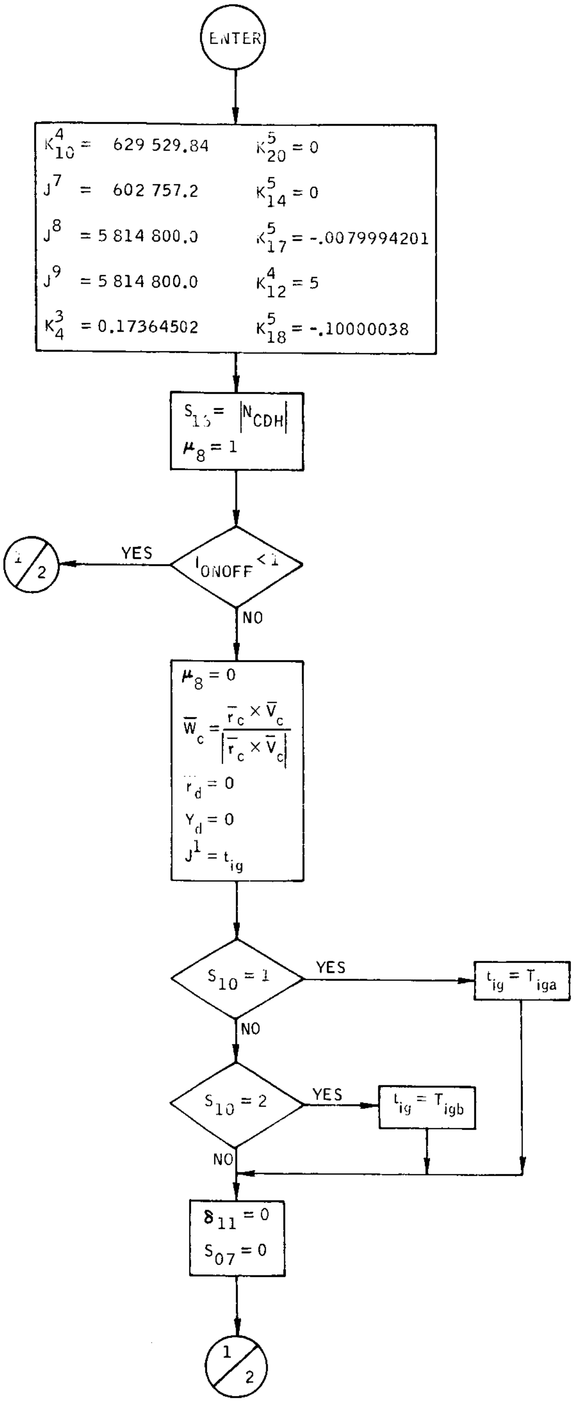
Page 5 of 6

Figure B-2.- Continued.


8
CALL TRANCO TO ROTATE X_{LSE} INTO SELENOCENTRIC COORDINATES AT T_{LO} OBTAINING X_E OF THE LM
CALL XESTAB TO STORE X_E
PRIMARY COMPUTER
PGNCS
AGS
AGS MONITORING
NO
YES
PGNCS MONITORING
NO
YES
CALL XESTAB TO STORE X_E IN THE AGS COMPUTER
CALL XESTAB TO STORE X_E IN THE PGNCS COMPUTER
$I_{STATE} = 0$
EXIT

Page 6 of 6

Figure B-2.- Concluded.



Page 1 of 13

Figure B-3.- Flow chart of subroutine AGS.

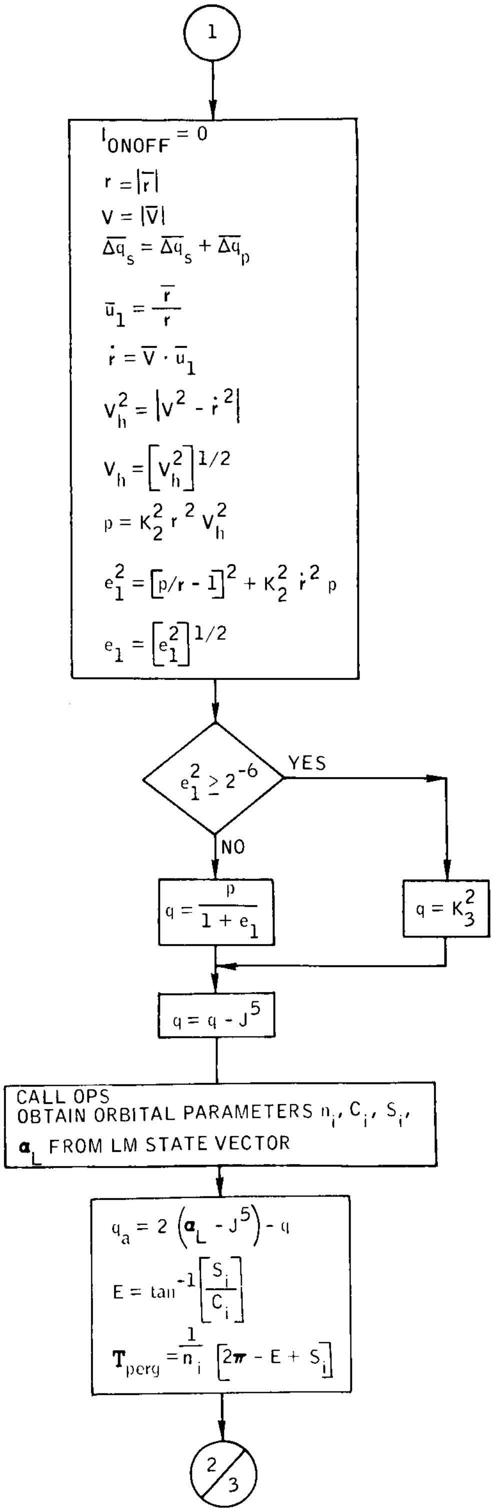
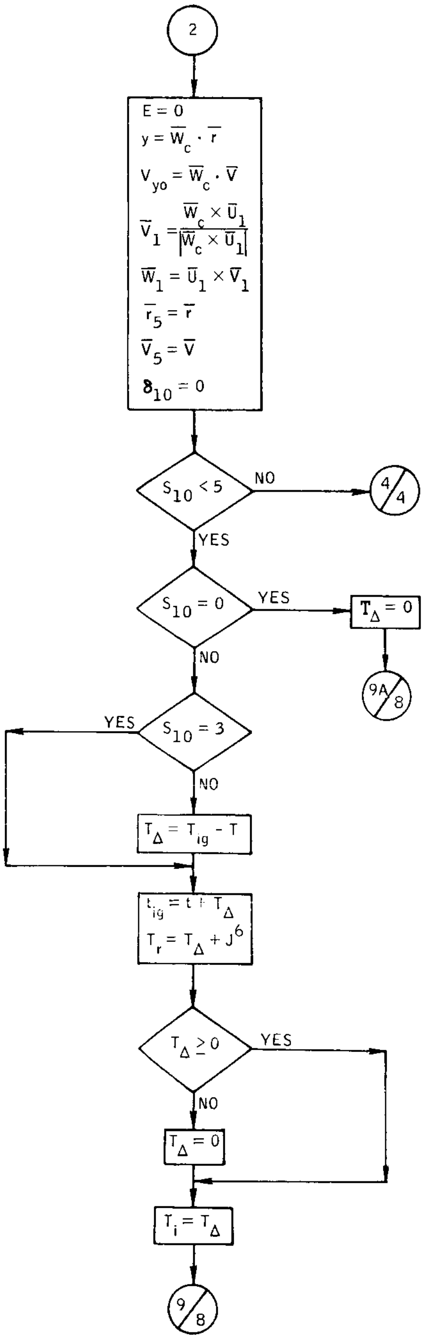
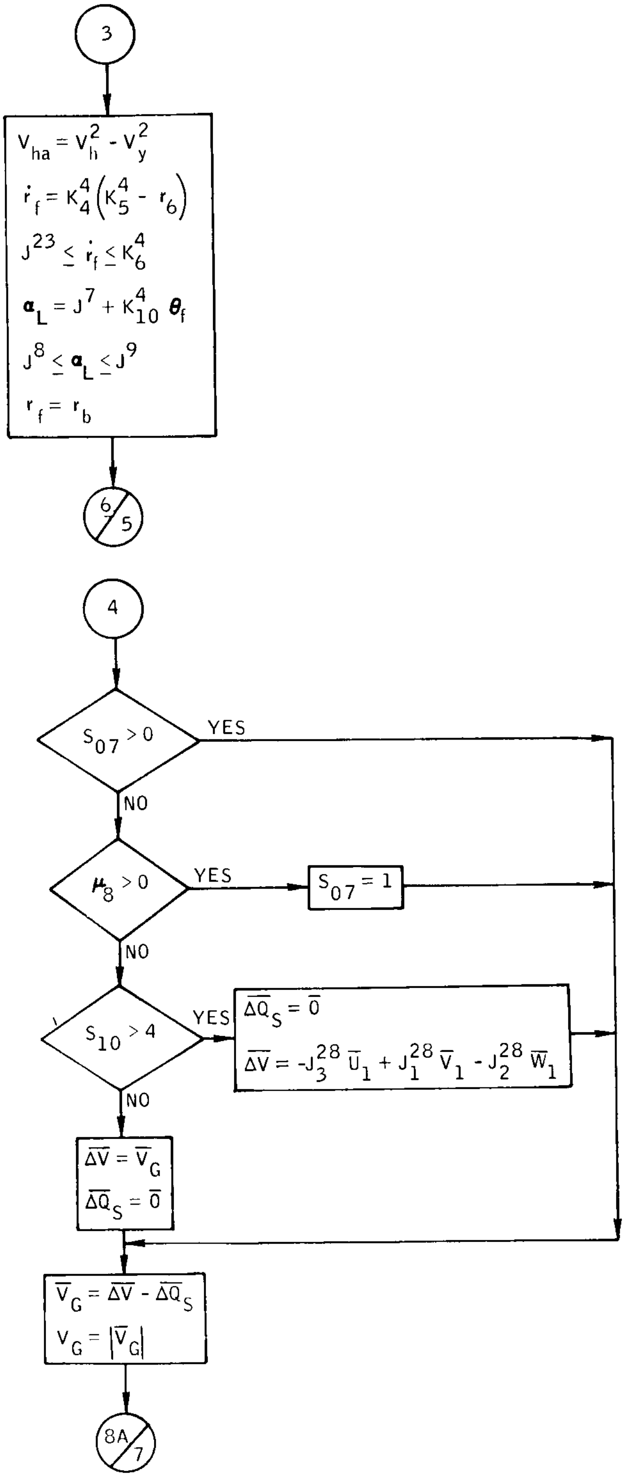

1
$I_{ONOFF} = 0$
$r = |\bar{r}|$
$V = |\bar{V}|$
$\overline{\Delta q}_s = \overline{\Delta q}_s + \overline{\Delta q}_p$
$\bar{u}_1 = \frac{\bar{r}}{r}$
$\dot{r} = \bar{V} \cdot \bar{u}_1$
$V_h^2 = |V^2 - \dot{r}^2|$
$V_h = \left[V_h^2\right]^{1/2}$
$p = K_2^2 r^2 V_h^2$
$e_1^2 = \left[p/r - 1\right]^2 + K_2^2 \dot{r}^2 p$
$e_1 = \left[e_1^2\right]^{1/2}$
$e_1^2 \geq 2^{-6}$
YES
NO
$q = \frac{p}{1 + e_1}$
$q = K_3^2$
$q = q - J^5$
CALL OPS
OBTAIN ORBITAL PARAMETERS n_i, C_i, S_i,
α_L FROM LM STATE VECTOR
$q_a = 2\left(\alpha_L - J^5\right) - q$
$E = \tan^{-1}\left[\frac{S_i}{C_i}\right]$
$T_{perg} = \frac{1}{n_i}\left[2\pi - E + S_i\right]$
2/3

Figure B-3.- Continued.


2
$E = 0$
$y = \overline{W}_c \cdot \overline{r}$
$V_{yo} = \overline{W}_c \cdot \overline{V}$
$\overline{V}_1 = \frac{\overline{W}_c \times \overline{U}_1}{|\overline{W}_c \times \overline{U}_1|}$
$\overline{W}_1 = \overline{U}_1 \times \overline{V}_1$
$\overline{r}_5 = \overline{r}$
$\overline{V}_5 = \overline{V}$
$\delta_{10} = 0$
$S_{10} < 5$
NO
4/4
YES
$S_{10} = 0$
YES
$T_\Delta = 0$
9A/8
NO
$S_{10} = 3$
YES
NO
$T_\Delta = T_{ig} - T$
$t_{ig} = t + T_\Delta$
$T_r = T_\Delta + J^6$
$T_\Delta \geq 0$
YES
NO
$T_\Delta = 0$
$T_i = T_\Delta$
9/8

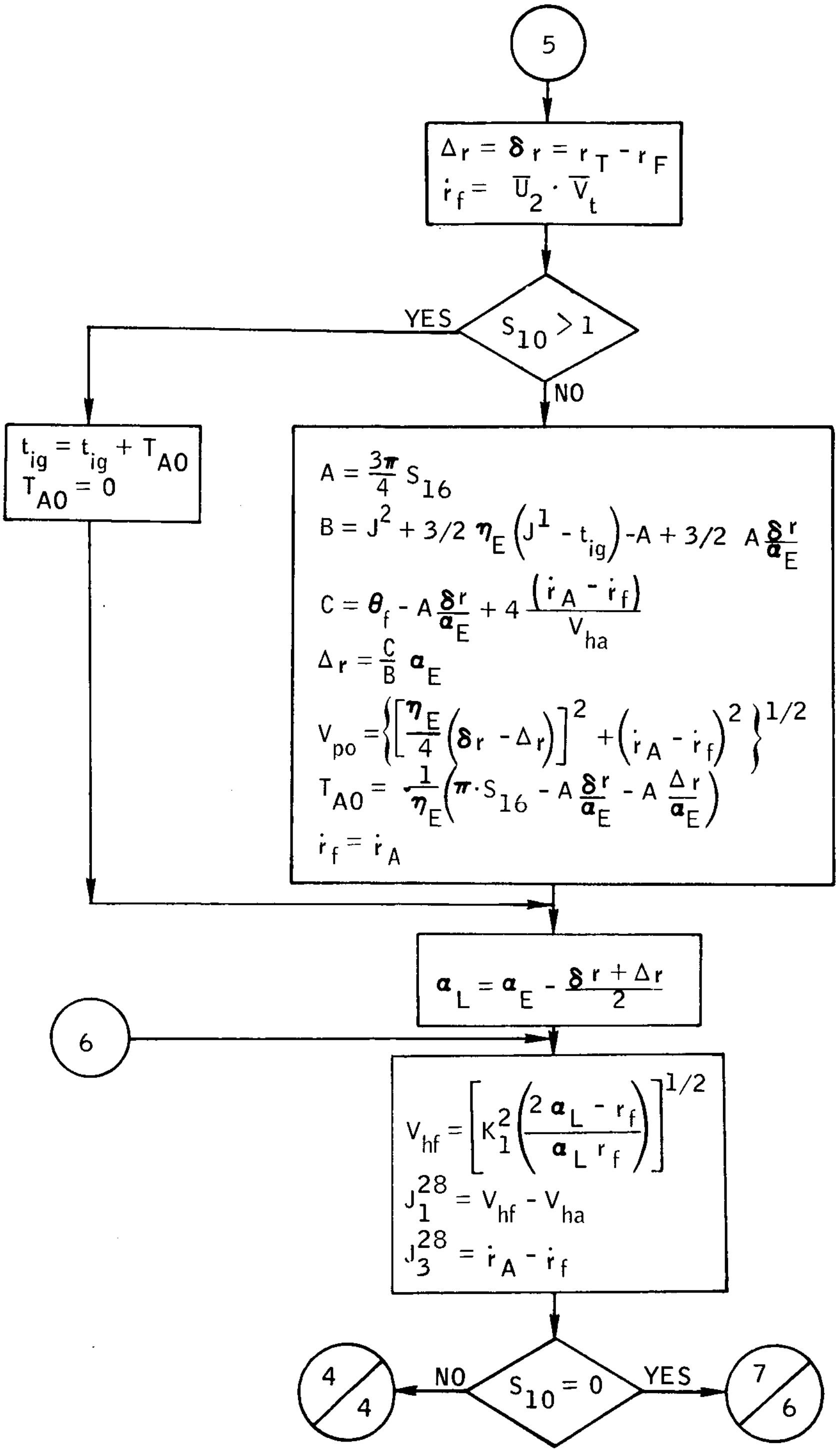
Page 3 of 13

Figure B-3.- Continued.


3
$V_{ha} = V_h^2 - V_y^2$
$\dot{r}_f = K_4^4 \left(K_5^4 - r_6\right)$
$J^{23} \leq \dot{r}_f \leq K_6^4$
$\alpha_L = J^7 + K_{10}^4 \theta_f$
$J^8 \leq \alpha_L \leq J^9$
$r_f = r_b$
6/5
4
$S_{07} > 0$
YES
NO
$\mu_8 > 0$
YES
$S_{07} = 1$
NO
$S_{10} > 4$
YES
$\overline{\Delta Q}_S = \overline{0}$
$\overline{\Delta V} = -J_3^{28} \overline{U}_1 + J_1^{28} \overline{V}_1 - J_2^{28} \overline{W}_1$
NO
$\overline{\Delta V} = \overline{V}_G$
$\overline{\Delta Q}_S = \overline{0}$
$\overline{V}_G = \overline{\Delta V} - \overline{\Delta Q}_S$
$V_G = \left|\overline{V}_G\right|$
8A/7

Page 4 of 13

Figure B-3.- Continued.


5
$\Delta_r = \delta_r = r_T - r_F$
$\dot{r}_f = \overline{U}_2 \cdot \overline{V}_t$
YES
$S_{10} > 1$
NO
$t_{ig} = t_{ig} + T_{A0}$
$T_{A0} = 0$
$A = \frac{3\pi}{4} S_{16}$
$B = J^2 + 3/2\ \eta_E \left(J^1 - t_{ig}\right) - A + 3/2\ A \frac{\delta_r}{\alpha_E}$
$C = \theta_f - A \frac{\delta_r}{\alpha_E} + 4 \frac{\left(\dot{r}_A - \dot{r}_f\right)}{V_{ha}}$
$\Delta_r = \frac{C}{B} \alpha_E$
$V_{po} = \left\{\left[\frac{\eta_E}{4}\left(\delta_r - \Delta_r\right)\right]^2 + \left(\dot{r}_A - \dot{r}_f\right)^2\right\}^{1/2}$
$T_{A0} = \frac{1}{\eta_E}\left(\pi \cdot S_{16} - A \frac{\delta_r}{\alpha_E} - A \frac{\Delta_r}{\alpha_E}\right)$
$\dot{r}_f = \dot{r}_A$
$\alpha_L = \alpha_E - \frac{\delta_r + \Delta_r}{2}$
6
$V_{hf} = \left[K_1^2 \left(\frac{2\alpha_L - r_f}{\alpha_L r_f}\right)\right]^{1/2}$
$J_1^{28} = V_{hf} - V_{ha}$
$J_3^{28} = \dot{r}_A - \dot{r}_f$
NO
$S_{10} = 0$
YES
4/4
7/6

Page 5 of 13

Figure B-3.- Continued.

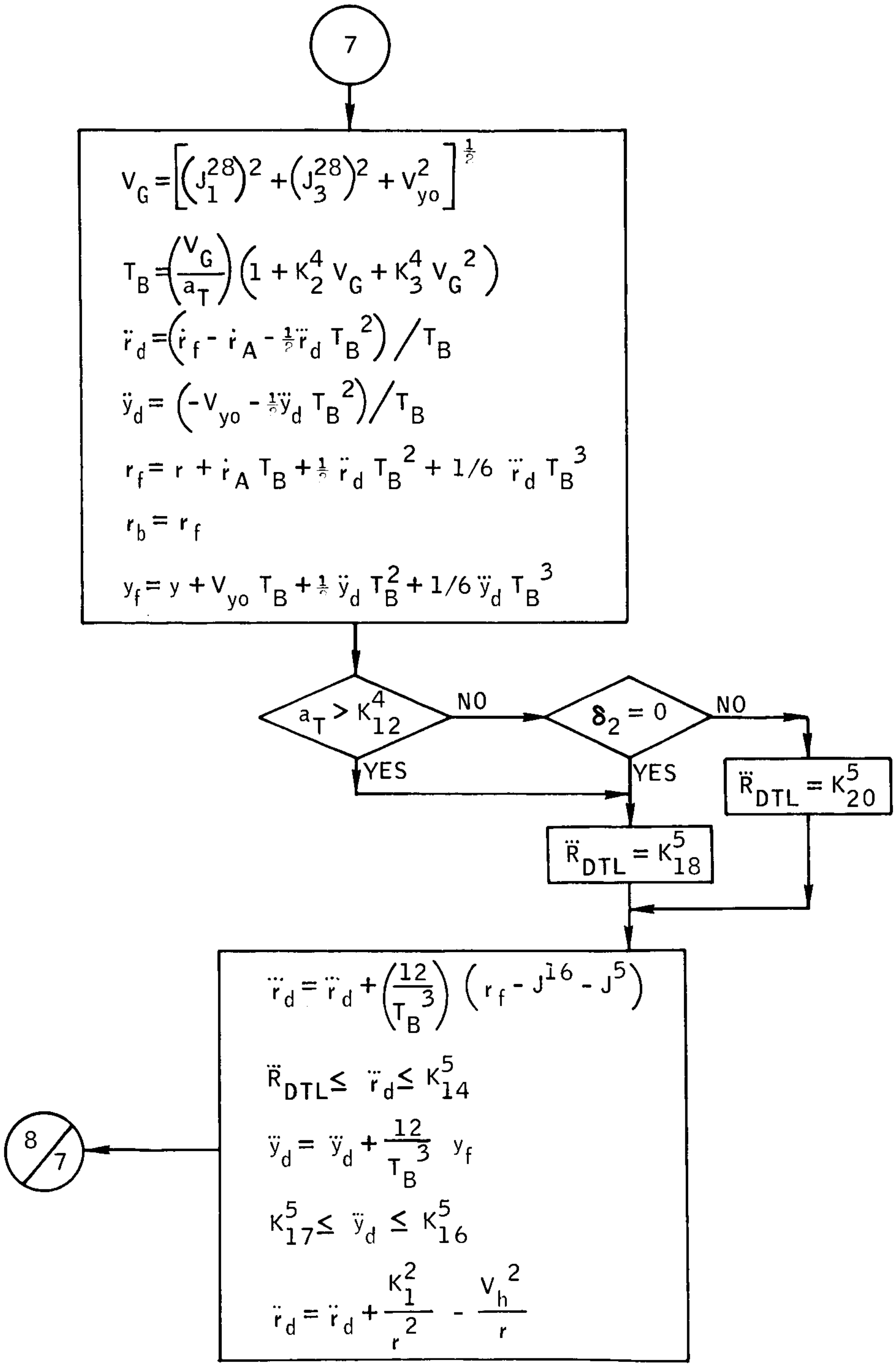
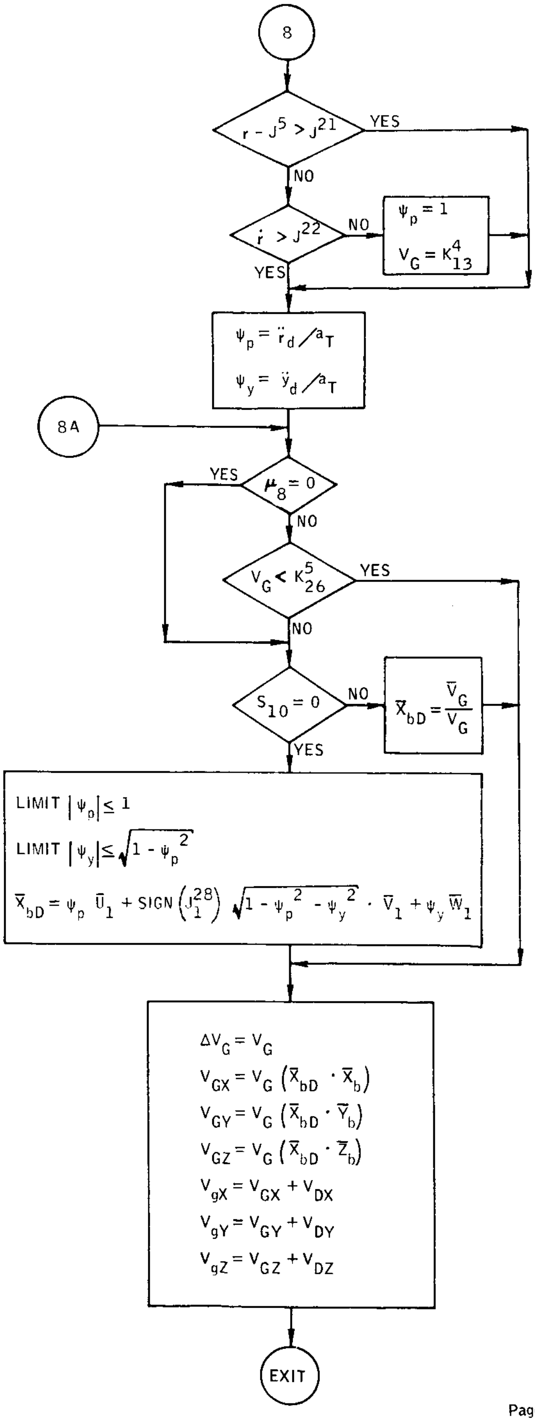

7
$V_G = \left[\left(J_1^{28}\right)^2 + \left(J_3^{28}\right)^2 + V_{yo}^2\right]^{\frac{1}{2}}$
$T_B = \left(\frac{V_G}{a_T}\right)\left(1 + K_2^4 V_G + K_3^4 V_G{}^2\right)$
$\ddot{r}_d = \left(\dot{r}_f - \dot{r}_A - \frac{1}{2}\dddot{r}_d T_B{}^2\right) / T_B$
$\ddot{y}_d = \left(-V_{yo} - \frac{1}{2}\dddot{y}_d T_B{}^2\right) / T_B$
$r_f = r + \dot{r}_A T_B + \frac{1}{2} \ddot{r}_d T_B{}^2 + 1/6 \dddot{r}_d T_B{}^3$
$r_b = r_f$
$y_f = y + V_{yo} T_B + \frac{1}{2} \ddot{y}_d T_B^2 + 1/6 \ddot{y}_d T_B{}^3$
$a_T > K_{12}^4$
NO
YES
$\delta_2 = 0$
NO
YES
$\dddot{R}_{DTL} = K_{20}^5$
$\dddot{R}_{DTL} = K_{18}^5$
$\dddot{r}_d = \dddot{r}_d + \left(\frac{12}{T_B{}^3}\right)\left(r_f - J^{16} - J^5\right)$
$\dddot{R}_{DTL} \leq \dddot{r}_d \leq K_{14}^5$
$\dddot{y}_d = \dddot{y}_d + \frac{12}{T_B{}^3} \; y_f$
$K_{17}^5 \leq \dddot{y}_d \leq K_{16}^5$
$\ddot{r}_d = \ddot{r}_d + \frac{K_1^2}{r^2} - \frac{V_h{}^2}{r}$
8/7

Figure B-3.- Continued.


8
$r - J^5 > J^{21}$
YES
NO
$\dot{r} > J^{22}$
NO
$\psi_p = 1$
$V_G = K^4_{13}$
YES
$\psi_p = \ddot{r}_d / a_T$
$\psi_y = \ddot{y}_d / a_T$
8A
YES
$\mu_8 = 0$
NO
$V_G < K^5_{26}$
YES
NO
$S_{10} = 0$
NO
$\bar{X}_{bD} = \frac{\bar{V}_G}{V_G}$
YES
LIMIT $|\psi_p| \leq 1$
LIMIT $|\psi_y| \leq \sqrt{1 - \psi_p^2}$
$\bar{X}_{bD} = \psi_p \bar{U}_1 + \text{SIGN}\left(J^{28}_1\right) \sqrt{1 - \psi_p^2 - \psi_y^2} \cdot \bar{V}_1 + \psi_y \bar{W}_1$
$\Delta V_G = V_G$
$V_{GX} = V_G \left(\bar{X}_{bD} \cdot \bar{X}_b\right)$
$V_{GY} = V_G \left(\bar{X}_{bD} \cdot \bar{Y}_b\right)$
$V_{GZ} = V_G \left(\bar{X}_{bD} \cdot \bar{Z}_b\right)$
$V_{gX} = V_{GX} + V_{DX}$
$V_{gY} = V_{GY} + V_{DY}$
$V_{gZ} = V_{GZ} + V_{DZ}$
EXIT

Page 7 of 13

Figure B-3.- Continued.

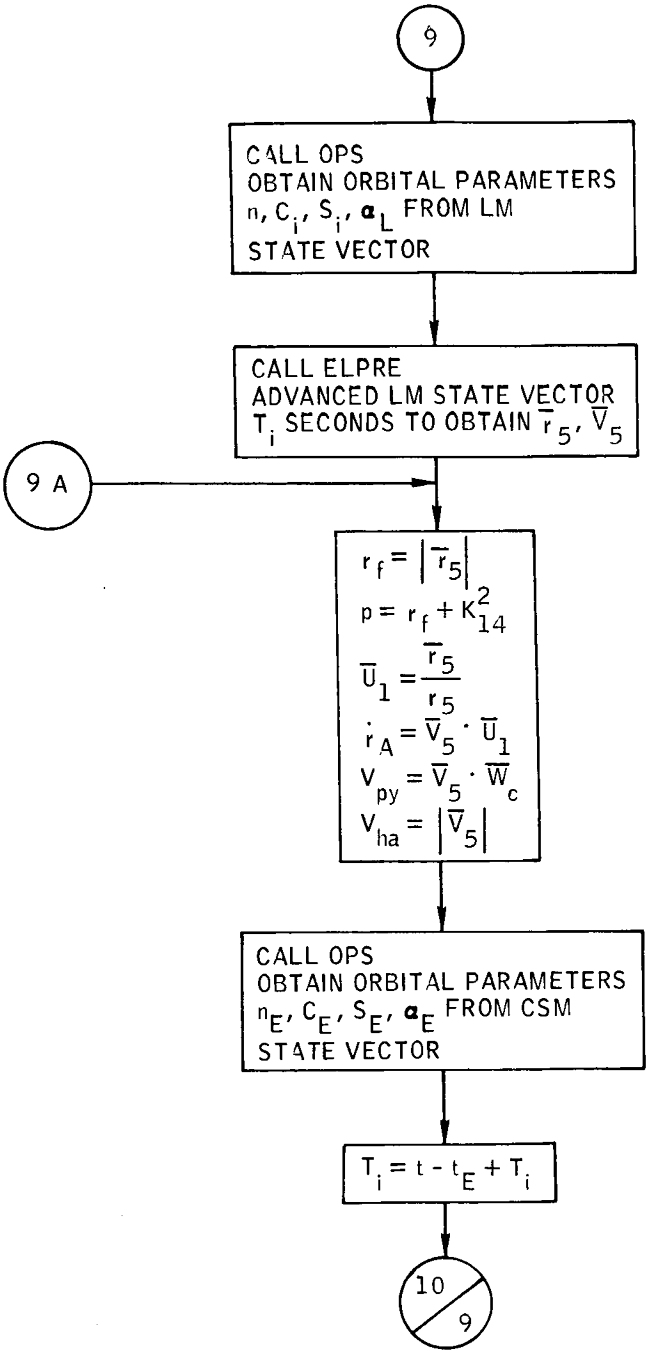

9
CALL OPS
OBTAIN ORBITAL PARAMETERS
n, C_i, S_i, α_L FROM LM
STATE VECTOR
CALL ELPRE
ADVANCED LM STATE VECTOR
T_i SECONDS TO OBTAIN $\overline{r}_5, \overline{V}_5$
9 A
$r_f = |\overline{r}_5|$
$p = r_f + K_{14}^2$
$\overline{U}_1 = \frac{\overline{r}_5}{r_5}$
$\dot{r}_A = \overline{V}_5 \cdot \overline{U}_1$
$V_{py} = \overline{V}_5 \cdot \overline{W}_c$
$V_{ha} = |\overline{V}_5|$
CALL OPS
OBTAIN ORBITAL PARAMETERS
n_E, C_E, S_E, α_E FROM CSM
STATE VECTOR
$T_i = t - t_E + T_i$
10
9

Figure B-3.- Continued.

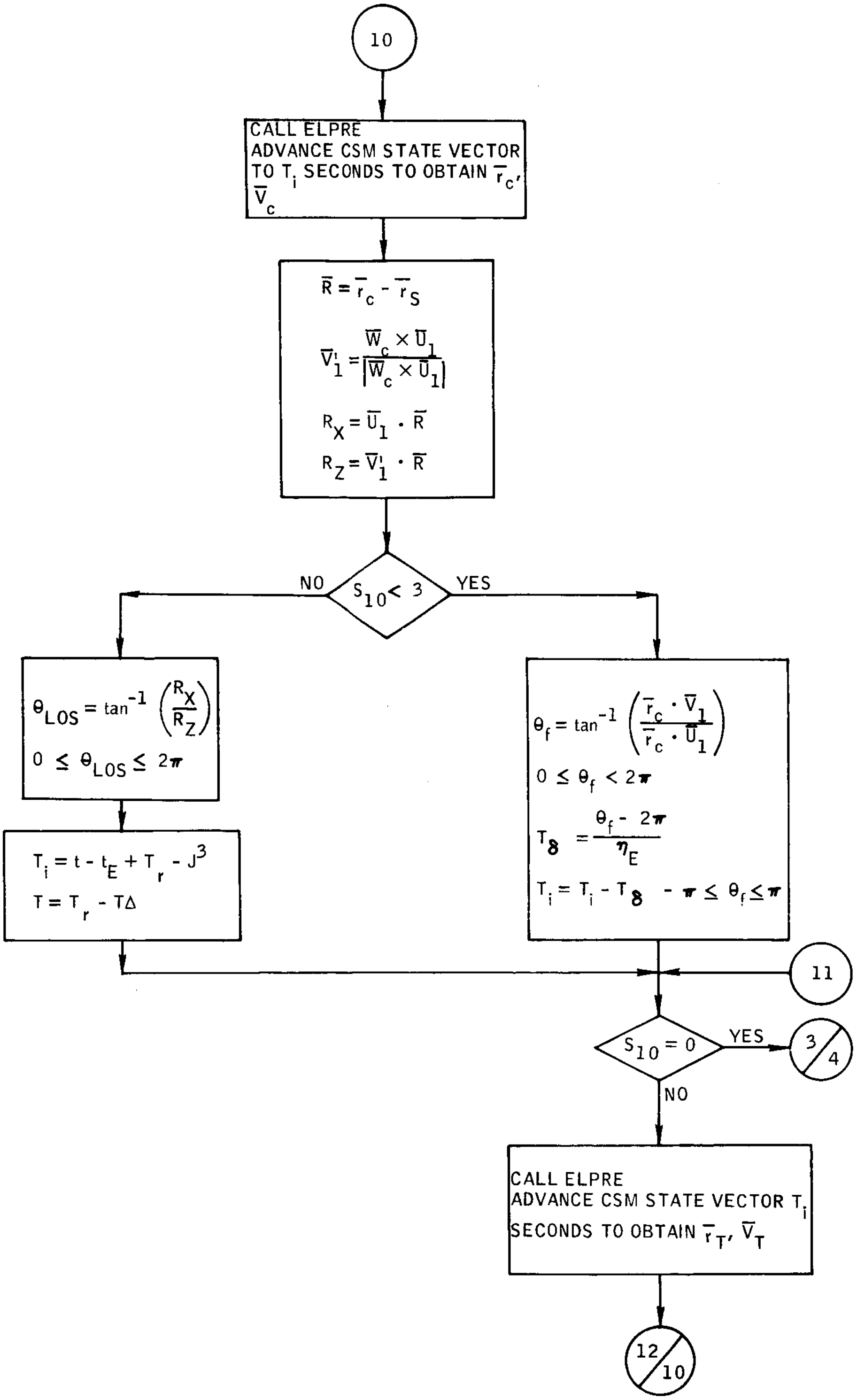
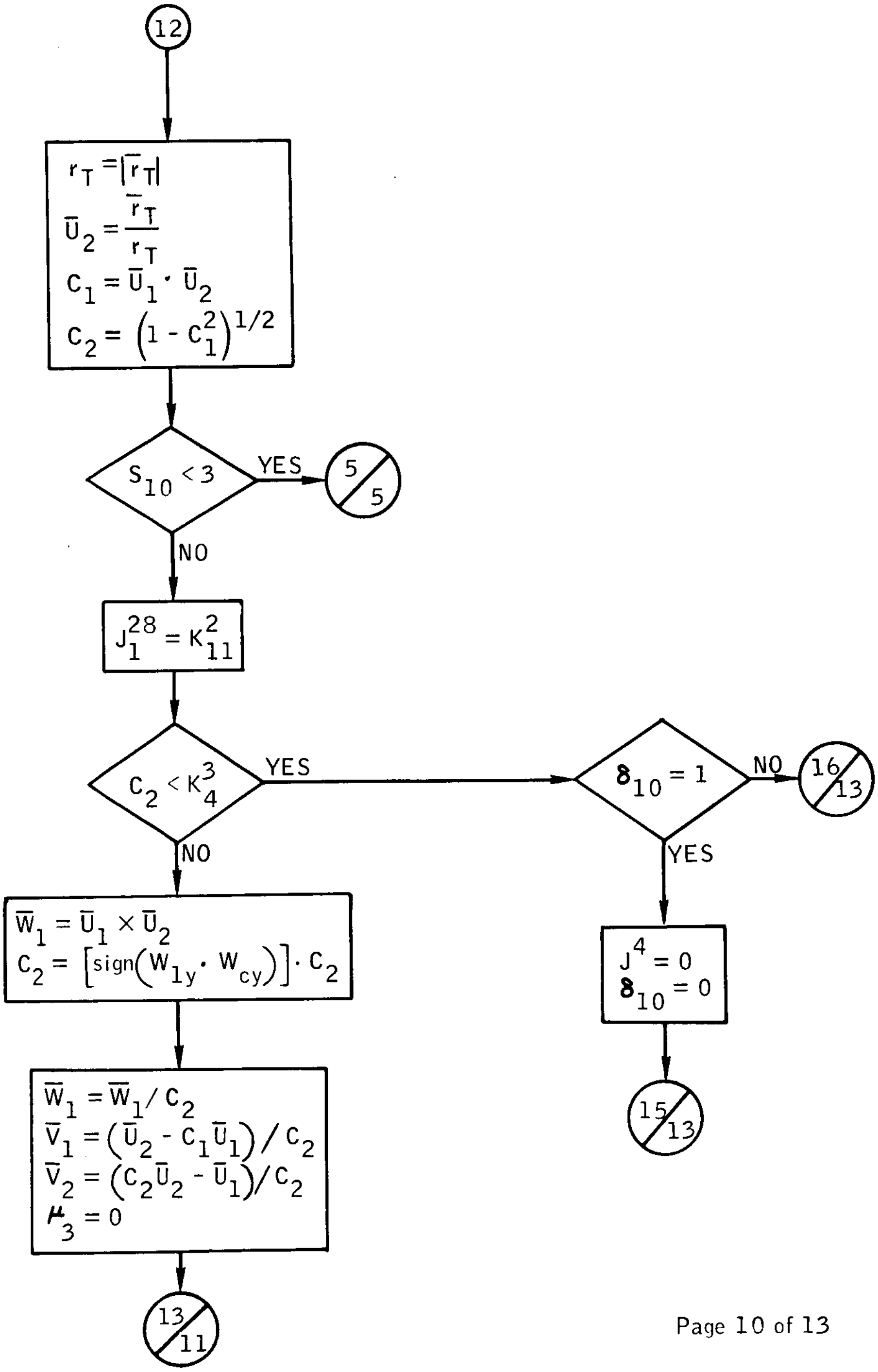

10
CALL ELPRE
ADVANCE CSM STATE VECTOR TO T_i SECONDS TO OBTAIN $\bar{r}_c$, $\bar{V}_c$
$\bar{R} = \bar{r}_c - \bar{r}_S$
$\bar{V}'_1 = \frac{\bar{W}_c \times \bar{U}_1}{|\bar{W}_c \times \bar{U}_1|}$
$R_X = \bar{U}_1 \cdot \bar{R}$
$R_Z = \bar{V}'_1 \cdot \bar{R}$
NO
$S_{10} < 3$
YES
$\theta_{LOS} = \tan^{-1}\left(\frac{R_X}{R_Z}\right)$
$0 \leq \theta_{LOS} \leq 2\pi$
$T_i = t - t_E + T_r - J^3$
$T = T_r - T\Delta$
$\theta_f = \tan^{-1}\left(\frac{\bar{r}_c \cdot \bar{V}_1}{\bar{r}_c \cdot \bar{U}_1}\right)$
$0 \leq \theta_f < 2\pi$
$T_\delta = \frac{\theta_f - 2\pi}{\eta_E}$
$T_i = T_i - T_\delta \quad -\pi \leq \theta_f \leq \pi$
11
$S_{10} = 0$
YES
3/4
NO
CALL ELPRE
ADVANCE CSM STATE VECTOR T_i SECONDS TO OBTAIN $\bar{r}_T$, $\bar{V}_T$
12/10

Figure B-3.- Continued.


12
$r_T = |\bar{r}_T|$
$\bar{U}_2 = \frac{\bar{r}_T}{r_T}$
$C_1 = \bar{U}_1 \cdot \bar{U}_2$
$C_2 = \left(1 - C_1^2\right)^{1/2}$
$S_{10} < 3$
YES
5/5
NO
$J_1^{28} = K_{11}^2$
$C_2 < K_4^3$
YES
$\delta_{10} = 1$
NO
16/13
YES
$J^4 = 0$
$\delta_{10} = 0$
15/13
NO
$\bar{W}_1 = \bar{U}_1 \times \bar{U}_2$
$C_2 = \left[\text{sign}\left(W_{1y} \cdot W_{cy}\right)\right] \cdot C_2$
$\bar{W}_1 = \bar{W}_1 / C_2$
$\bar{V}_1 = \left(\bar{U}_2 - C_1\bar{U}_1\right) / C_2$
$\bar{V}_2 = \left(C_2\bar{U}_2 - \bar{U}_1\right) / C_2$
$\mu_3 = 0$
13/11

Page 10 of 13

Figure B-3.- Continued.

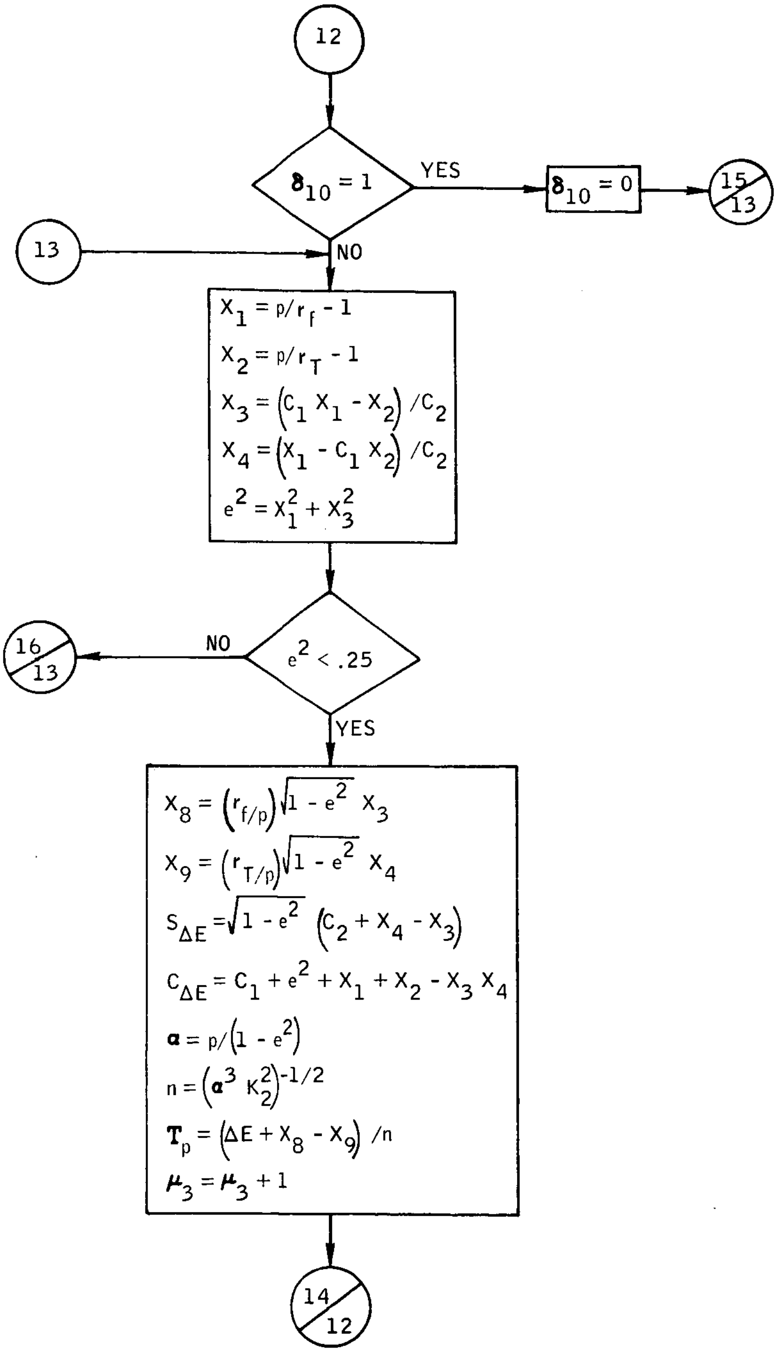

12
$\delta_{10} = 1$
YES
$\delta_{10} = 0$
15/13
13
NO
$X_1 = p/r_f - 1$
$X_2 = p/r_T - 1$
$X_3 = \left(C_1 X_1 - X_2\right) / C_2$
$X_4 = \left(X_1 - C_1 X_2\right) / C_2$
$e^2 = X_1^2 + X_3^2$
16/13
NO
$e^2 < .25$
YES
$X_8 = \left(r_{f/p}\right)\sqrt{1 - e^2}\ X_3$
$X_9 = \left(r_{T/p}\right)\sqrt{1 - e^2}\ X_4$
$S_{\Delta E} = \sqrt{1 - e^2}\ \left(C_2 + X_4 - X_3\right)$
$C_{\Delta E} = C_1 + e^2 + X_1 + X_2 - X_3 X_4$
$\boldsymbol{a} = p/\left(1 - e^2\right)$
$n = \left(\boldsymbol{a}^3\ K_2^2\right)^{-1/2}$
$\mathbf{T}_p = \left(\Delta E + X_8 - X_9\right) / n$
$\boldsymbol{\mu}_3 = \boldsymbol{\mu}_3 + 1$
14/12

Figure .- Continued.

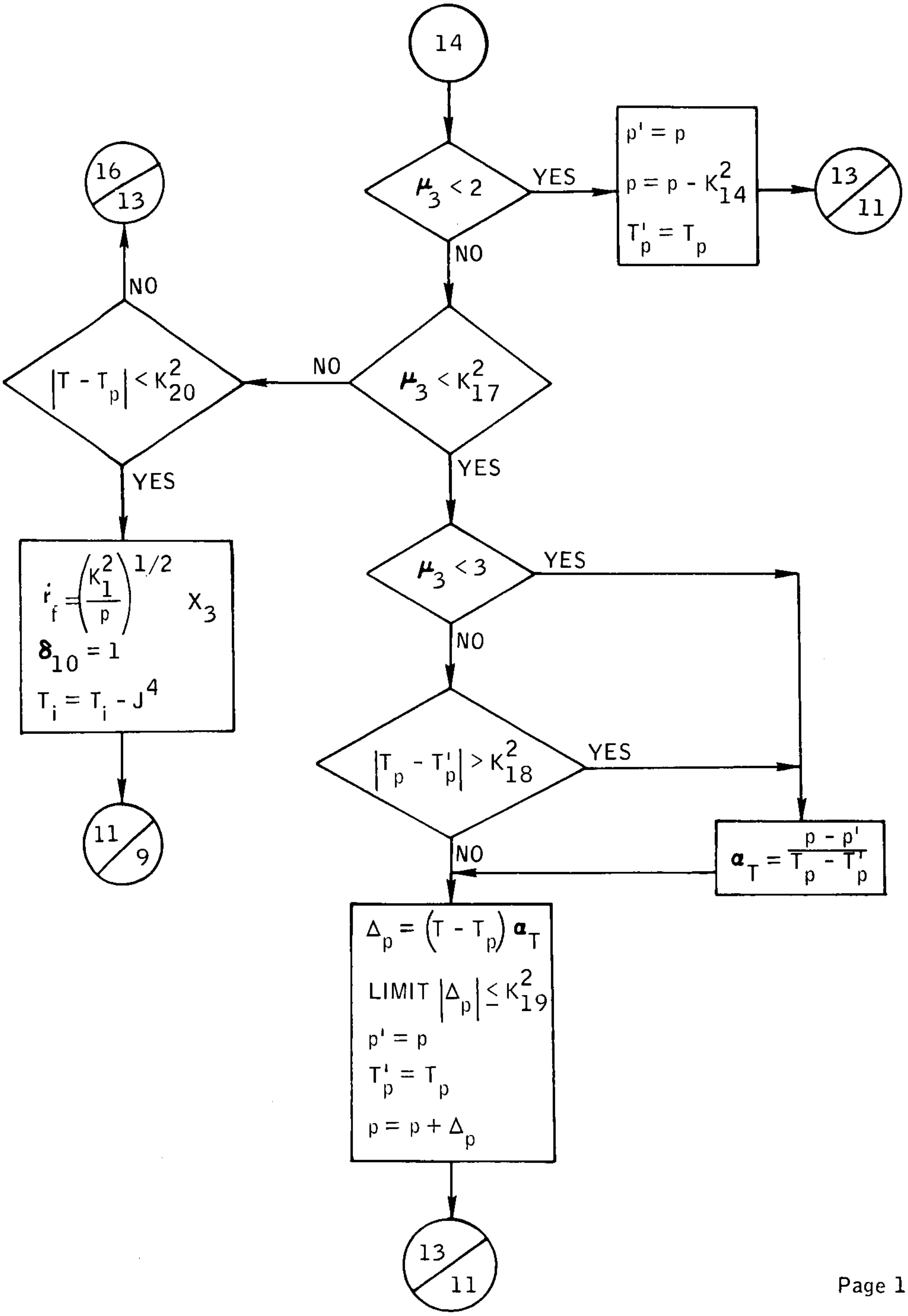
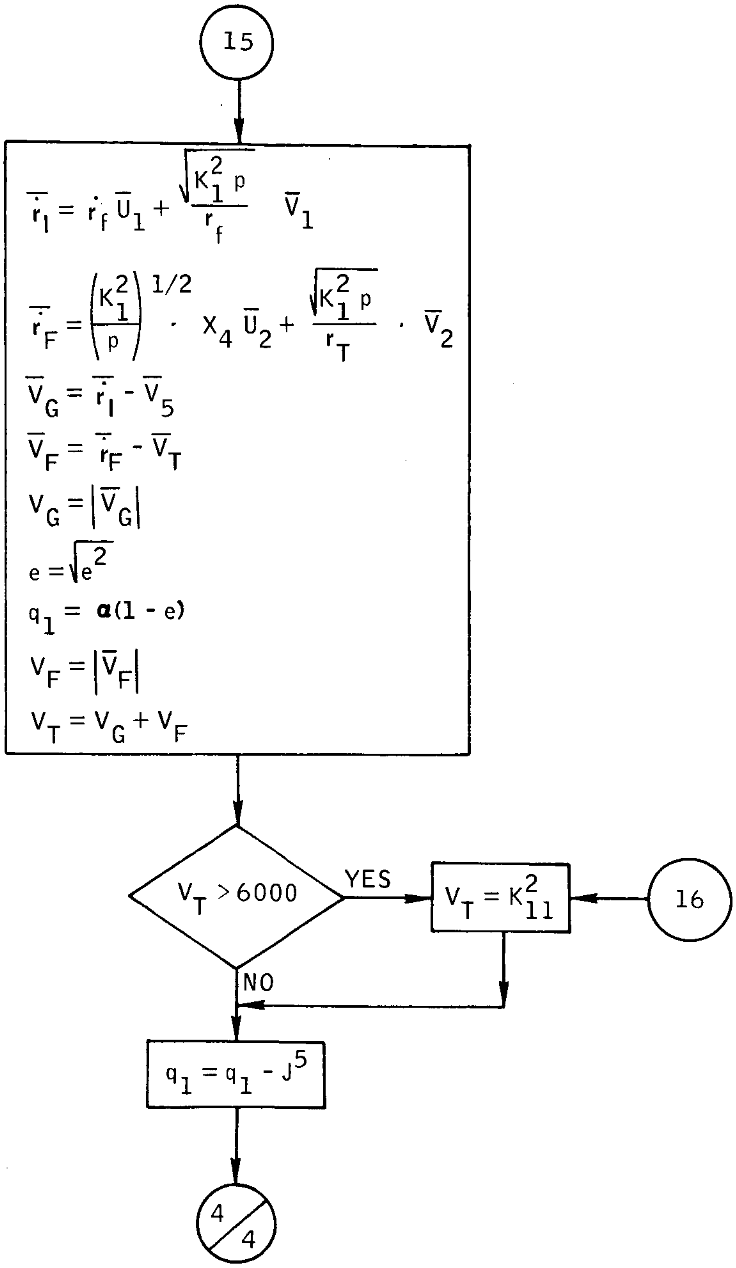


Figure B-3.- Continued.


15
$\bar{\dot{r}}_I = \dot{r}_f \bar{U}_1 + \sqrt{\frac{K_1^2 p}{r_f}} \bar{V}_1$
$\bar{\dot{r}}_F = \left(\frac{K_1^2}{p}\right)^{1/2} \cdot X_4 \bar{U}_2 + \sqrt{\frac{K_1^2 p}{r_T}} \cdot \bar{V}_2$
$\bar{V}_G = \bar{\dot{r}}_I - \bar{V}_5$
$\bar{V}_F = \bar{\dot{r}}_F - \bar{V}_T$
$V_G = |\bar{V}_G|$
$e = \sqrt{e^2}$
$q_1 = \alpha(1 - e)$
$V_F = |\bar{V}_F|$
$V_T = V_G + V_F$
$V_T > 6000$
YES
NO
$V_T = K_{11}^2$
16
$q_1 = q_1 - J^5$
4
4

Page 13 of 13

Figure B-3.- Concluded.

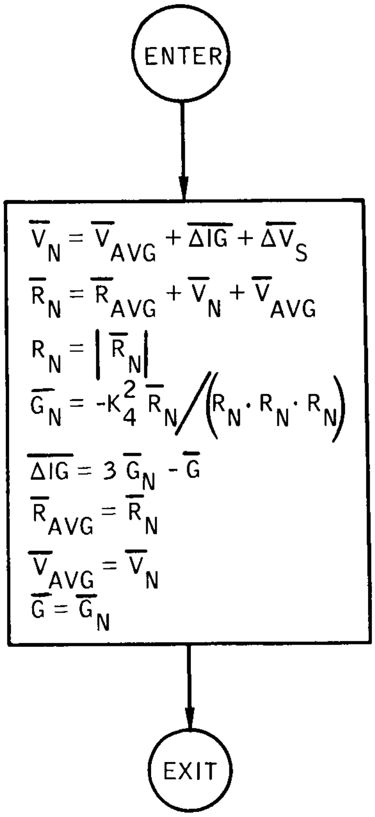

ENTER
$\overline{V}_N = \overline{V}_{AVG} + \overline{\Delta IG} + \overline{\Delta V}_S$
$\overline{R}_N = \overline{R}_{AVG} + \overline{V}_N + \overline{V}_{AVG}$
$R_N = \left| \overline{R}_N \right|$
$\overline{G}_N = -K_4^2 \overline{R}_N \Big/ \left(R_N \cdot R_N \cdot R_N \right)$
$\overline{\Delta IG} = 3 \, \overline{G}_N - \overline{G}$
$\overline{R}_{AVG} = \overline{R}_N$
$\overline{V}_{AVG} = \overline{V}_N$
$\overline{G} = \overline{G}_N$
EXIT

Figure B-4.- Flow chart of subroutine AGSAG. Page 1 of 1.

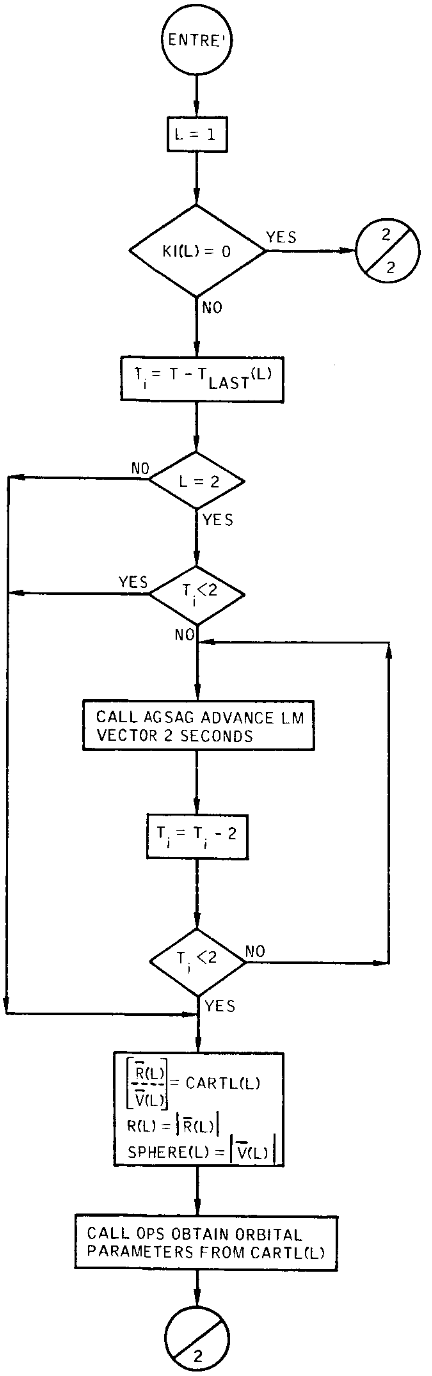

ENTRE'
L = 1
KI(L) = 0
YES
2
2
NO
$T_i = T - T_{LAST}(L)$
NO
L = 2
YES
YES
$T_i < 2$
NO
CALL AGSAG ADVANCE LM VECTOR 2 SECONDS
$T_i = T_i - 2$
$T_i < 2$
NO
YES
$\begin{bmatrix} \bar{R}(L) \\ \bar{V}(L) \end{bmatrix} = CARTL(L)$
$R(L) = |\bar{R}(L)|$
$SPHERE(L) = |\bar{V}(L)|$
CALL OPS OBTAIN ORBITAL PARAMETERS FROM CARTL(L)
2

Figure B-5.- Flow chart of subroutine AGSKEP.

Page 1 of 2

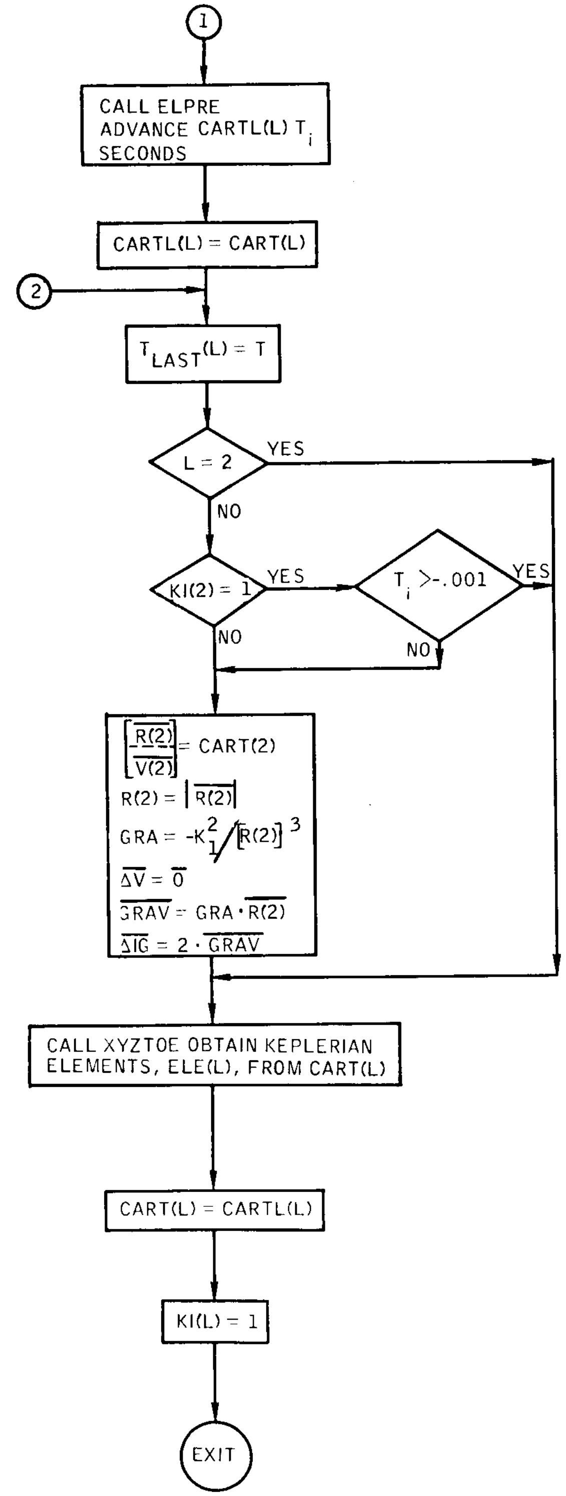
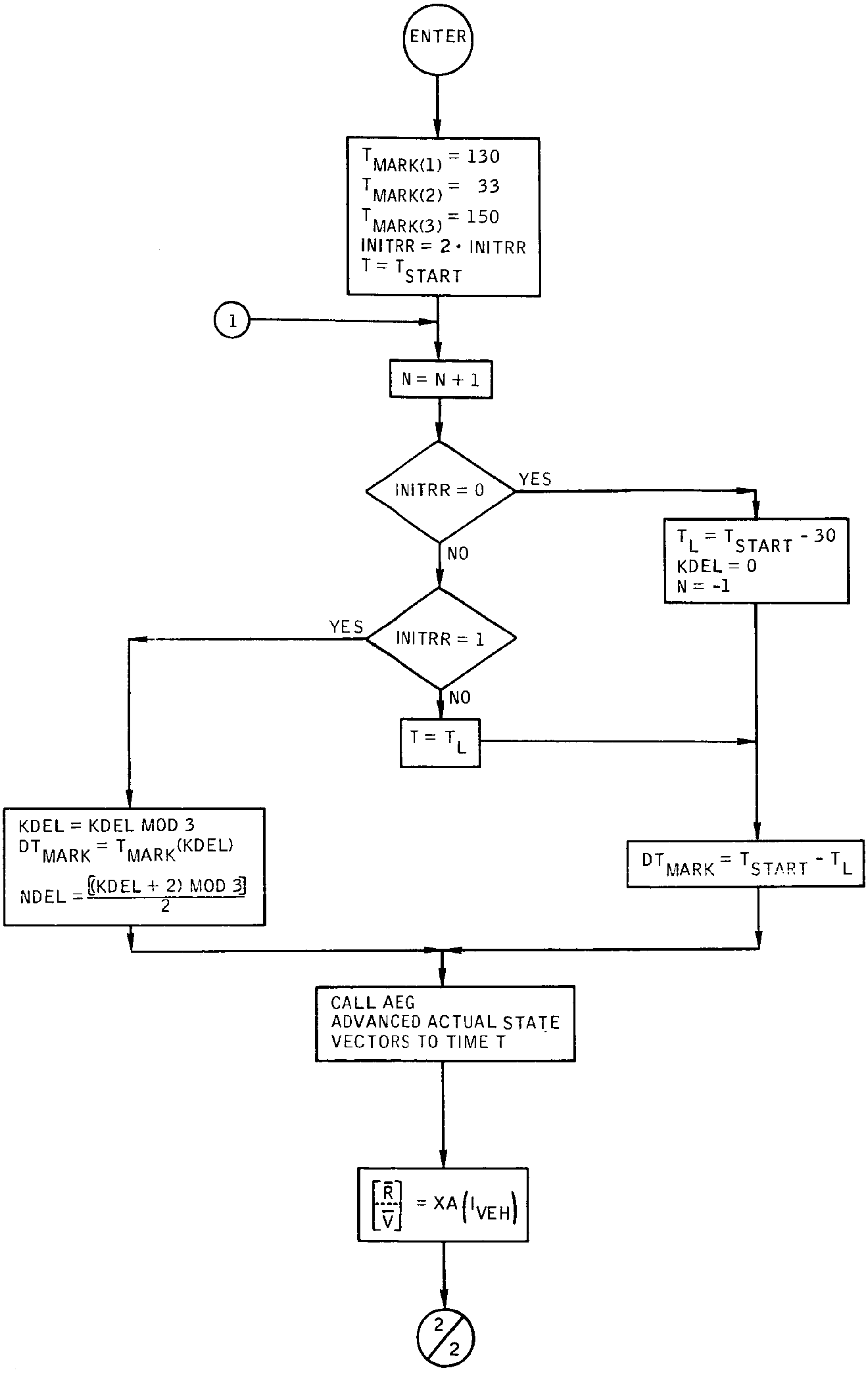

1
CALL ELPRE
ADVANCE CARTL(L) T_i
SECONDS
CARTL(L) = CART(L)
2
$T_{LAST}(L) = T$
L = 2
YES
NO
KI(2) = 1
YES
NO
T_i >-.001
YES
NO
$\begin{bmatrix}\overline{R(2)}\\\overline{V(2)}\end{bmatrix}$ = CART(2)
R(2) = $|\overline{R(2)}|$
GRA = $-K_1^2 / [R(2)]^3$
$\overline{\Delta V} = \overline{0}$
$\overline{GRAV}$ = GRA · $\overline{R(2)}$
$\overline{\Delta IG}$ = 2 · $\overline{GRAV}$
CALL XYZTOE OBTAIN KEPLERIAN
ELEMENTS, ELE(L), FROM CART(L)
CART(L) = CARTL(L)
KI(L) = 1
EXIT

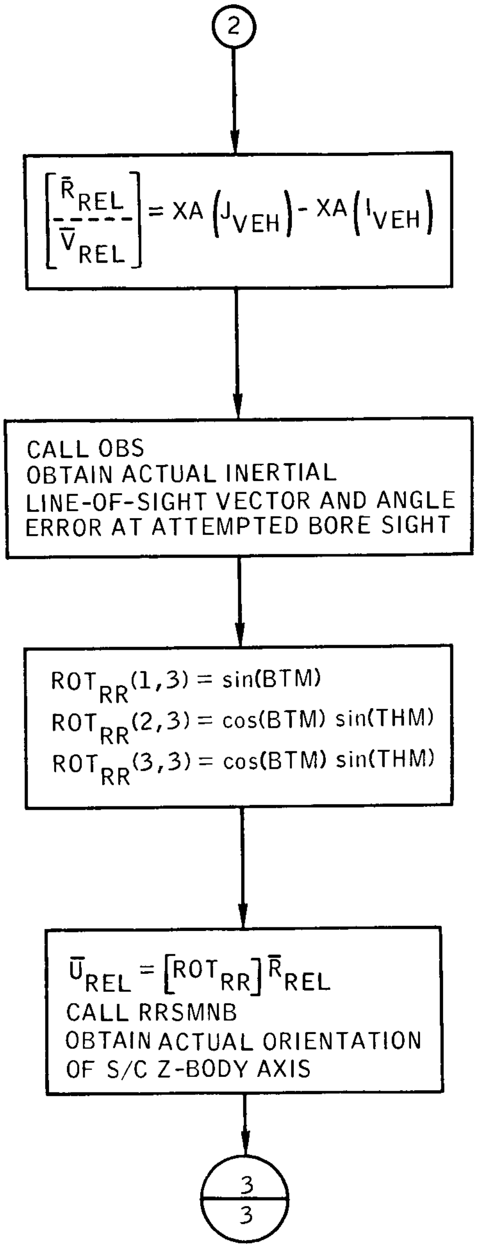
Figure B-5.- Concluded.

Page 2 of 2



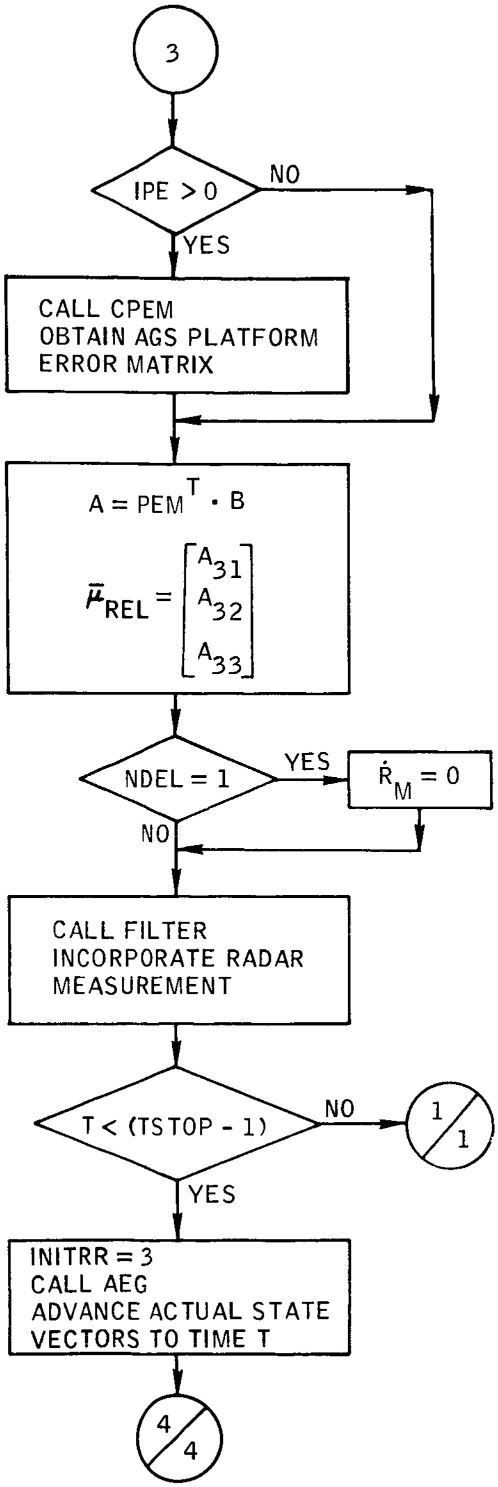
Page 1 of 4

Figure B-6.- Flow chart of subroutine AGSRR.


2
$\begin{bmatrix} \bar{R}_{REL} \\ \hline \bar{V}_{REL} \end{bmatrix} = XA\left(J_{VEH}\right) - XA\left(I_{VEH}\right)$
CALL OBS
OBTAIN ACTUAL INERTIAL
LINE-OF-SIGHT VECTOR AND ANGLE
ERROR AT ATTEMPTED BORE SIGHT
$ROT_{RR}(1,3) = \sin(BTM)$
$ROT_{RR}(2,3) = \cos(BTM) \sin(THM)$
$ROT_{RR}(3,3) = \cos(BTM) \sin(THM)$
$\bar{U}_{REL} = \left[ROT_{RR}\right] \bar{R}_{REL}$
CALL RRSMNB
OBTAIN ACTUAL ORIENTATION
OF S/C Z-BODY AXIS
3
3

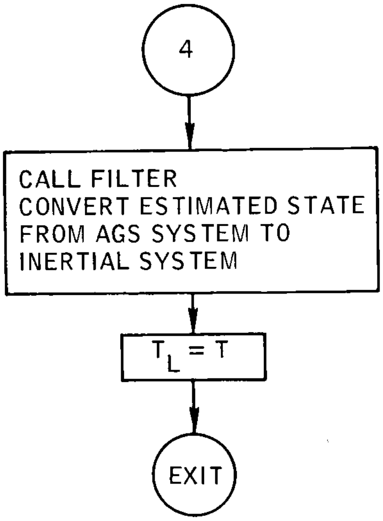
Page 2 of 4

Figure B-6.- Continued.



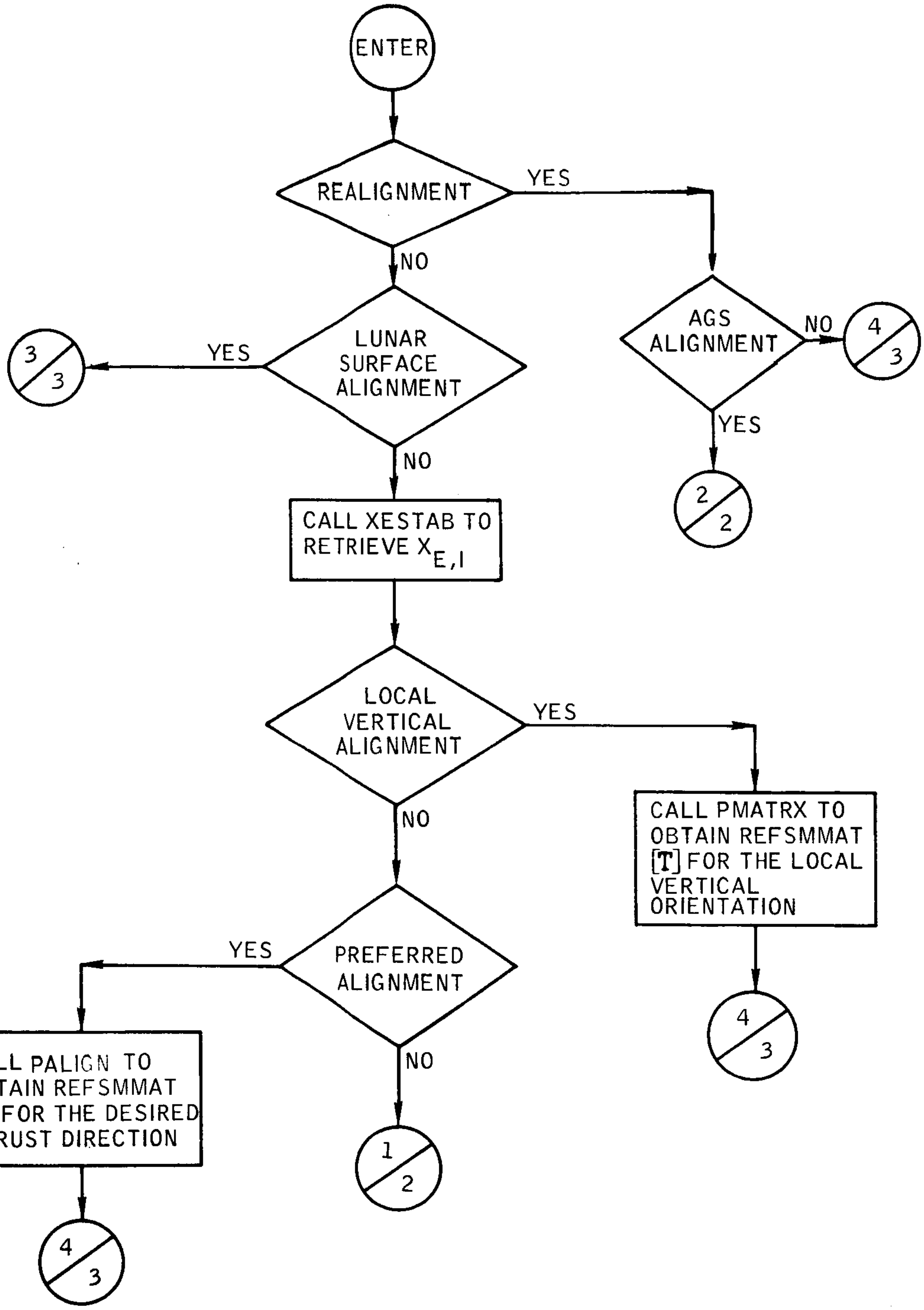
Page 3 of 4

Figure B-6.- Continued.



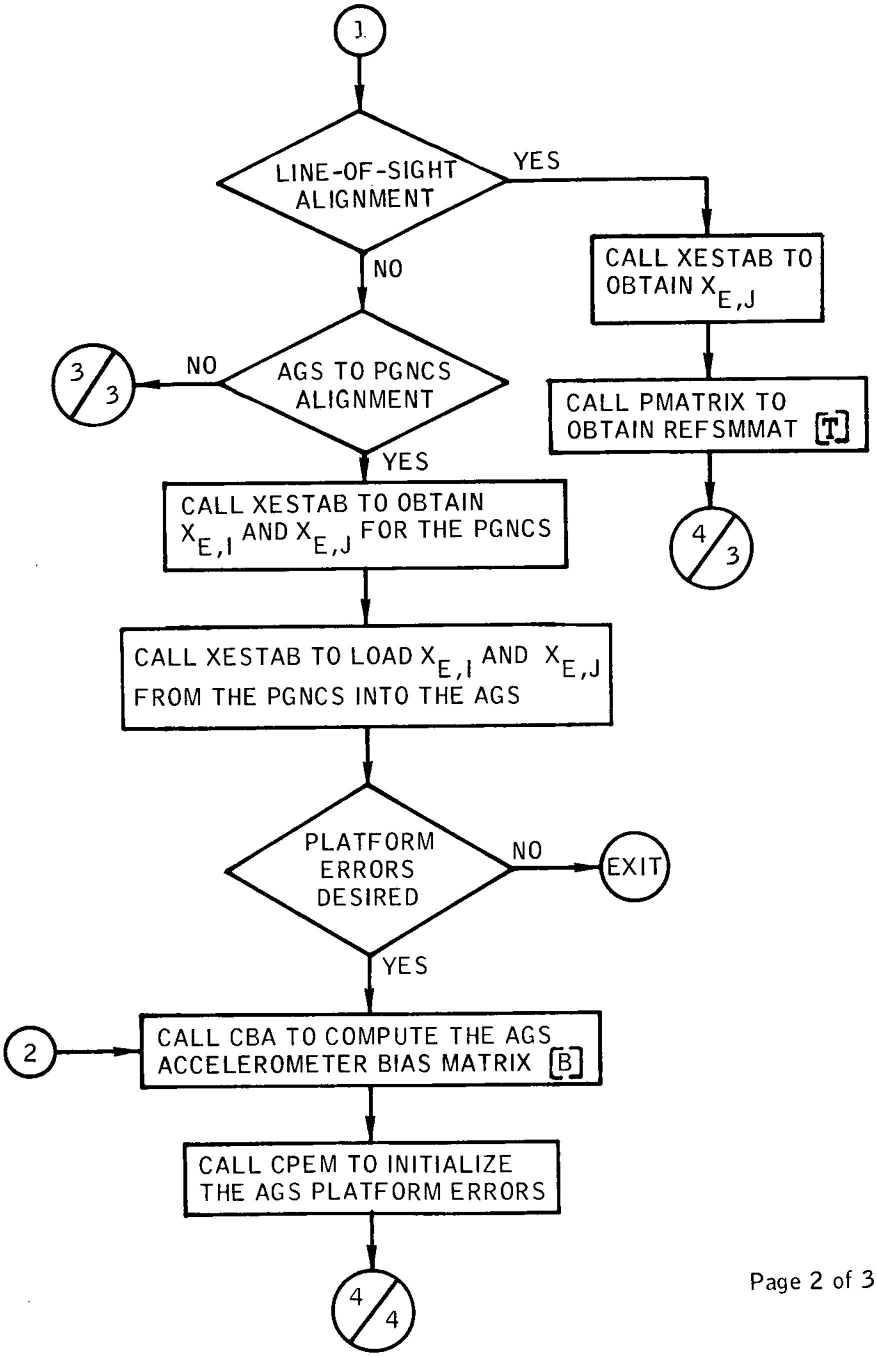
Page 4 of 4

Figure B-6.- Concluded.


ENTER
REALIGNMENT
YES
NO
AGS ALIGNMENT
NO
4/3
YES
2/2
LUNAR SURFACE ALIGNMENT
YES
3/3
NO
CALL XESTAB TO RETRIEVE $X_{E,I}$
LOCAL VERTICAL ALIGNMENT
YES
CALL PMATRX TO OBTAIN REFSMMAT [T] FOR THE LOCAL VERTICAL ORIENTATION
4/3
NO
PREFERRED ALIGNMENT
YES
CALL PALIGN TO OBTAIN REFSMMAT [T] FOR THE DESIRED THRUST DIRECTION
4/3
NO
1/2

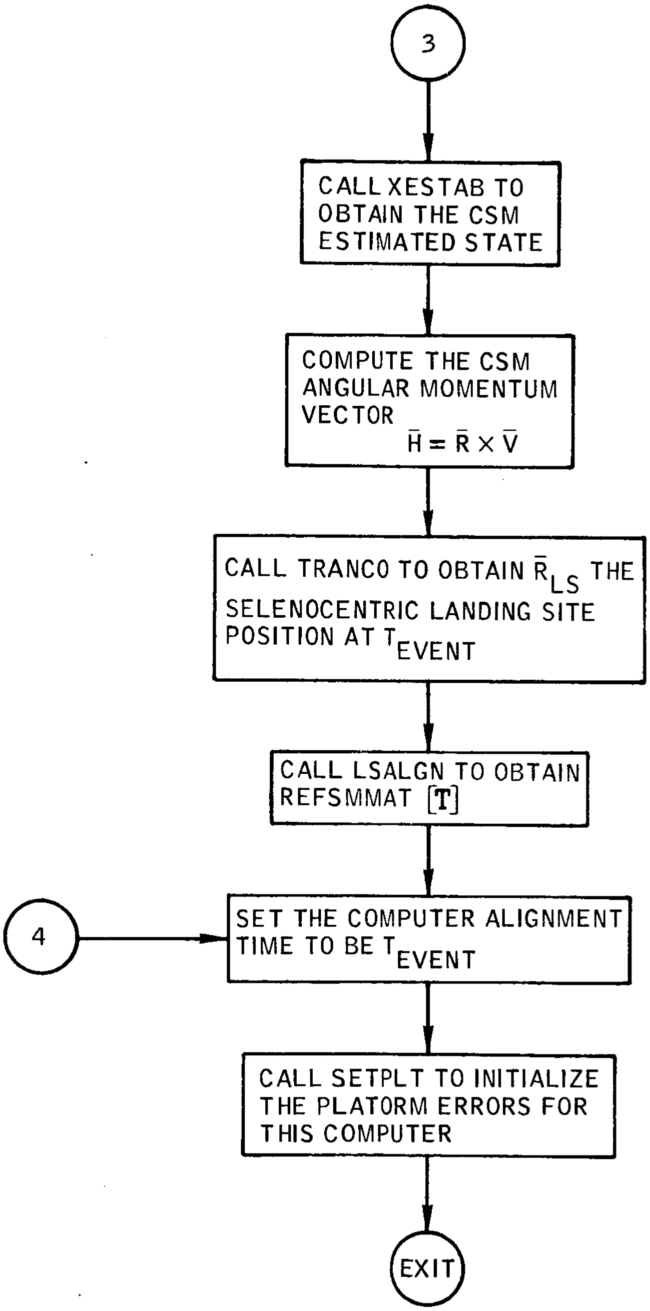
Page 1 of 3

Figure B-7.- Flow chart of subroutine ALIGN.


1
LINE-OF-SIGHT ALIGNMENT
YES
NO
CALL XESTAB TO OBTAIN $X_{E,J}$
CALL PMATRIX TO OBTAIN REFSMMAT [T]
4/3
AGS TO PGNCS ALIGNMENT
NO
3/3
YES
CALL XESTAB TO OBTAIN $X_{E,I}$ AND $X_{E,J}$ FOR THE PGNCS
CALL XESTAB TO LOAD $X_{E,I}$ AND $X_{E,J}$ FROM THE PGNCS INTO THE AGS
PLATFORM ERRORS DESIRED
NO
EXIT
YES
2
CALL CBA TO COMPUTE THE AGS ACCELEROMETER BIAS MATRIX [B]
CALL CPEM TO INITIALIZE THE AGS PLATFORM ERRORS
4/4

Page 2 of 3

Figure B-7.- Continued.


3
CALL XESTAB TO OBTAIN THE CSM ESTIMATED STATE
COMPUTE THE CSM ANGULAR MOMENTUM VECTOR
$\bar{H} = \bar{R} \times \bar{V}$
CALL TRANCO TO OBTAIN $\bar{R}_{LS}$ THE SELENOCENTRIC LANDING SITE POSITION AT T_{EVENT}
CALL LSALGN TO OBTAIN REFSMMAT $[T]$
4
SET THE COMPUTER ALIGNMENT TIME TO BE T_{EVENT}
CALL SETPLT TO INITIALIZE THE PLATORM ERRORS FOR THIS COMPUTER
EXIT

Page 3 of 3

Figure B-7.- Concluded.

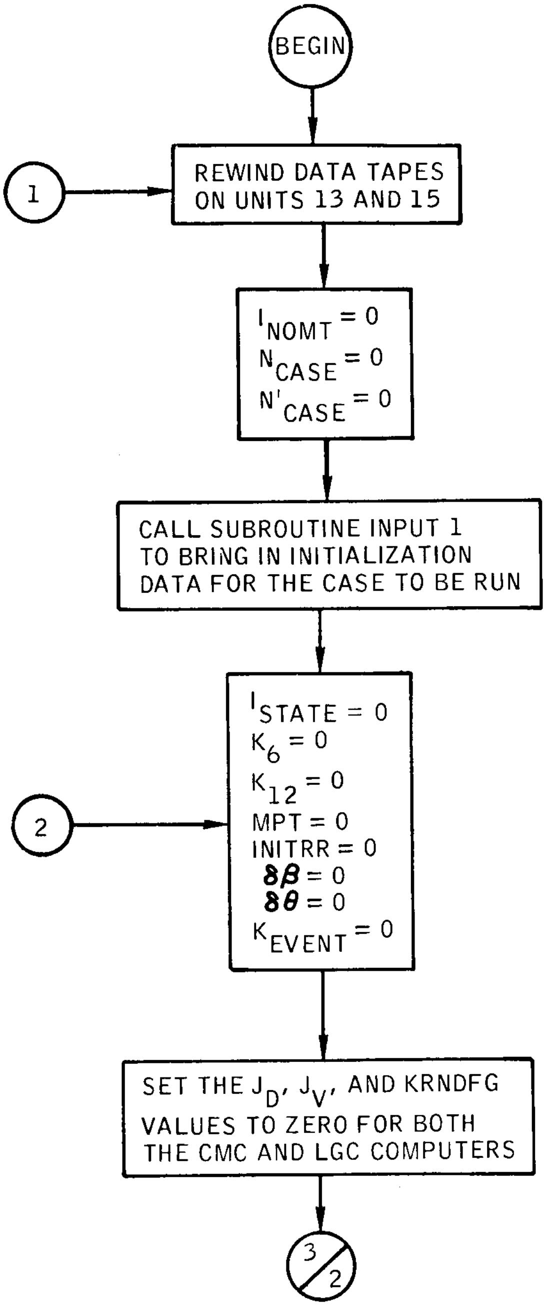
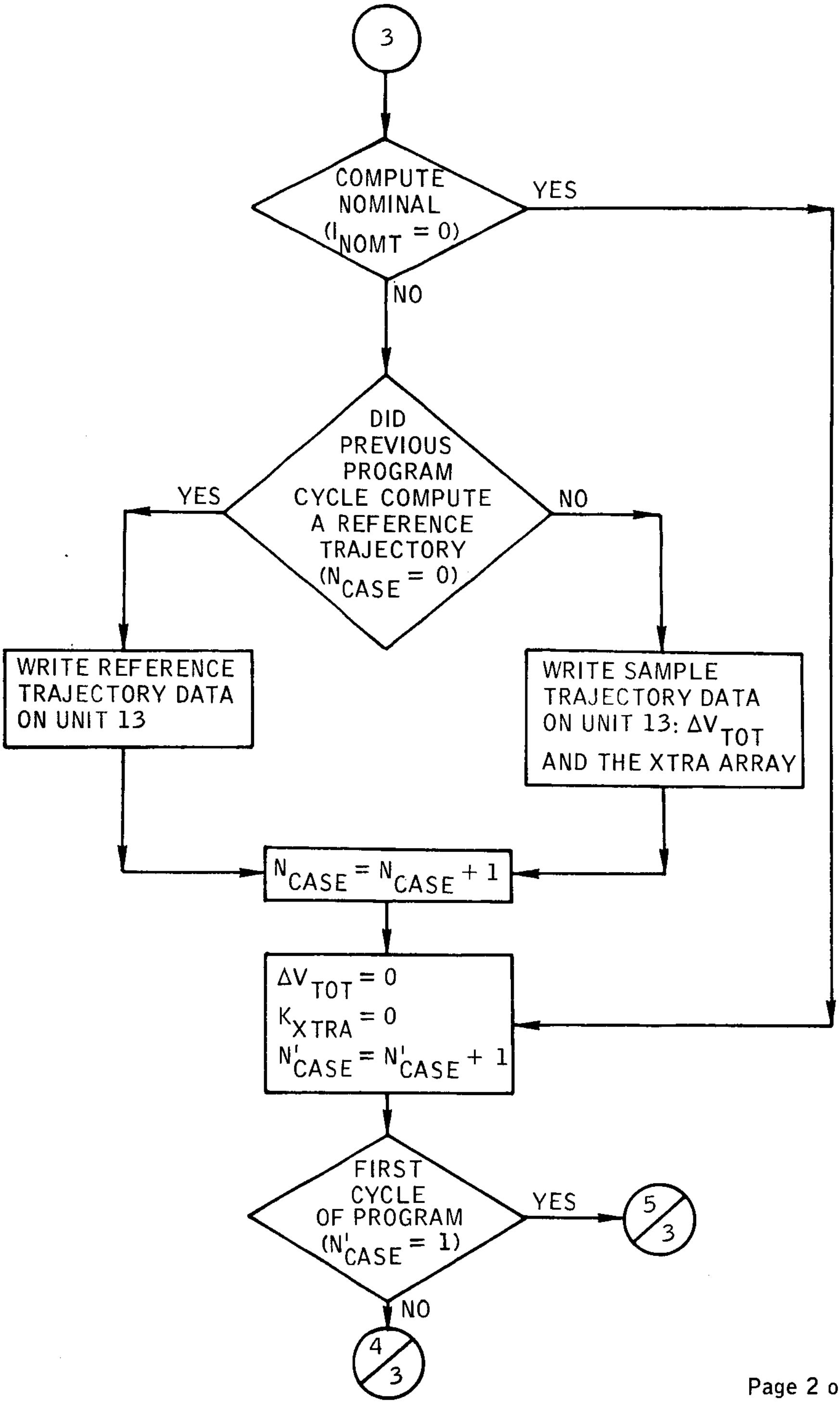
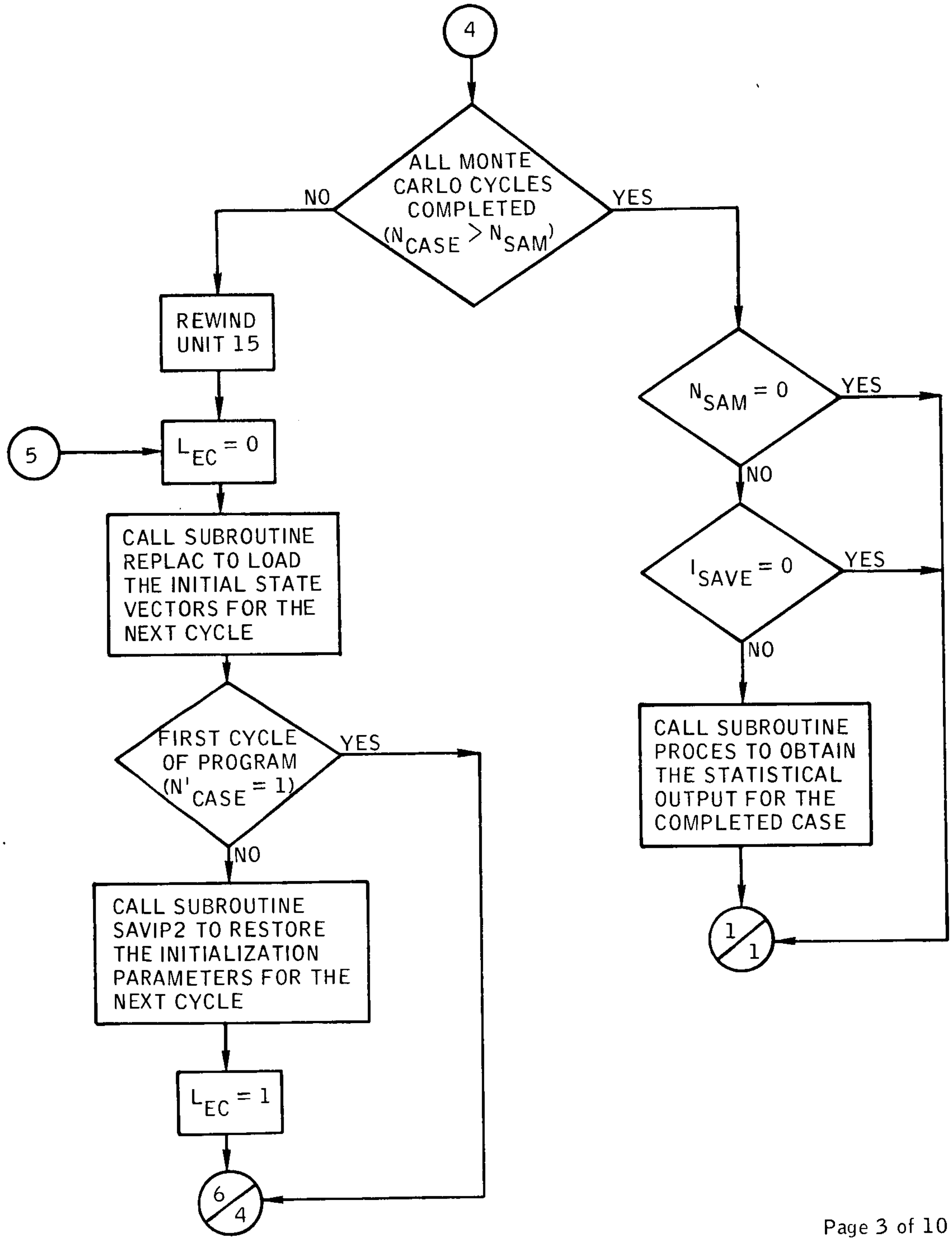

BEGIN
1
REWIND DATA TAPES ON UNITS 13 AND 15
$I_{NOMT} = 0$
$N_{CASE} = 0$
$N'_{CASE} = 0$
CALL SUBROUTINE INPUT 1 TO BRING IN INITIALIZATION DATA FOR THE CASE TO BE RUN
2
$I_{STATE} = 0$
$K_6 = 0$
$K_{12} = 0$
MPT = 0
INITRR = 0
$\delta\beta = 0$
$\delta\theta = 0$
$K_{EVENT} = 0$
SET THE J_D, J_V, AND KRNDFG VALUES TO ZERO FOR BOTH THE CMC AND LGC COMPUTERS
3
2

Figure B-8.- Flow chart of AGAST driver AMAIN. Page 1 of 10


3
COMPUTE NOMINAL (I_{NOMT} = 0)
YES
NO
DID PREVIOUS PROGRAM CYCLE COMPUTE A REFERENCE TRAJECTORY (N_{CASE} = 0)
YES
NO
WRITE REFERENCE TRAJECTORY DATA ON UNIT 13
WRITE SAMPLE TRAJECTORY DATA ON UNIT 13: ΔV_{TOT} AND THE XTRA ARRAY
$N_{CASE} = N_{CASE} + 1$
$\Delta V_{TOT} = 0$
$K_{XTRA} = 0$
$N'_{CASE} = N'_{CASE} + 1$
FIRST CYCLE OF PROGRAM (N'_{CASE} = 1)
YES
5/3
NO
4/3

Page 2 of 10

Figure B-8.- Continued.


4
ALL MONTE CARLO CYCLES COMPLETED ($N_{CASE} > N_{SAM}$)
NO
YES
REWIND UNIT 15
5
$L_{EC} = 0$
CALL SUBROUTINE REPLAC TO LOAD THE INITIAL STATE VECTORS FOR THE NEXT CYCLE
FIRST CYCLE OF PROGRAM ($N'_{CASE} = 1$)
YES
NO
CALL SUBROUTINE SAVIP2 TO RESTORE THE INITIALIZATION PARAMETERS FOR THE NEXT CYCLE
$L_{EC} = 1$
6/4
$N_{SAM} = 0$
YES
NO
$I_{SAVE} = 0$
YES
NO
CALL SUBROUTINE PROCES TO OBTAIN THE STATISTICAL OUTPUT FOR THE COMPLETED CASE
1/1

Page 3 of 10

Figure B-8.- Continued.

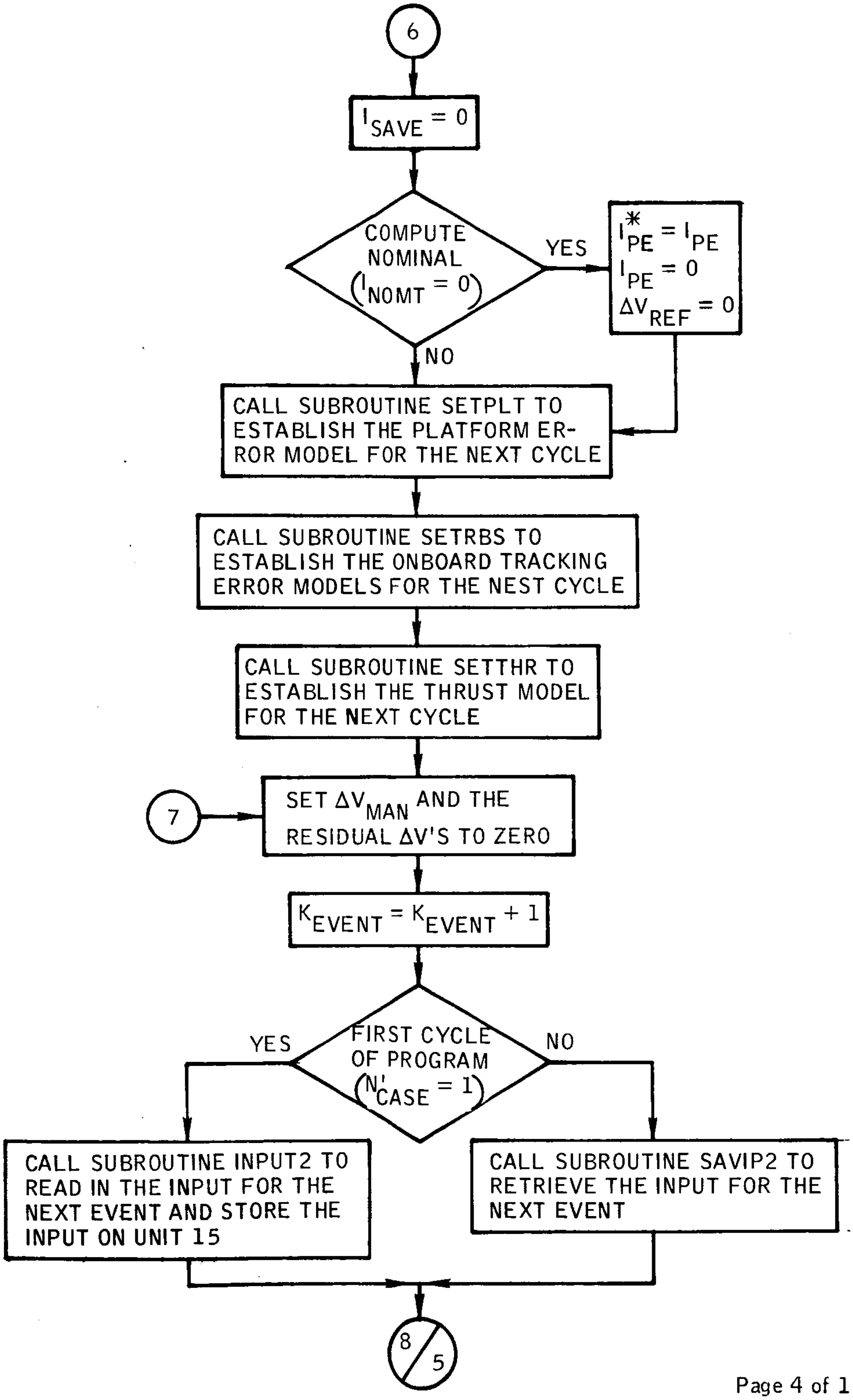

6
$I_{SAVE} = 0$
COMPUTE NOMINAL $(I_{NOMT} = 0)$
YES
$I^{*}_{PE} = I_{PE}$
$I_{PE} = 0$
$\Delta V_{REF} = 0$
NO
CALL SUBROUTINE SETPLT TO ESTABLISH THE PLATFORM ERROR MODEL FOR THE NEXT CYCLE
CALL SUBROUTINE SETRBS TO ESTABLISH THE ONBOARD TRACKING ERROR MODELS FOR THE NEST CYCLE
CALL SUBROUTINE SETTHR TO ESTABLISH THE THRUST MODEL FOR THE NEXT CYCLE
7
SET ΔV_{MAN} AND THE RESIDUAL ΔV'S TO ZERO
$K_{EVENT} = K_{EVENT} + 1$
FIRST CYCLE OF PROGRAM $(N'_{CASE} = 1)$
YES
NO
CALL SUBROUTINE INPUT2 TO READ IN THE INPUT FOR THE NEXT EVENT AND STORE THE INPUT ON UNIT 15
CALL SUBROUTINE SAVIP2 TO RETRIEVE THE INPUT FOR THE NEXT EVENT
8
5

Figure B-8.- Continued.

Page 4 of 10

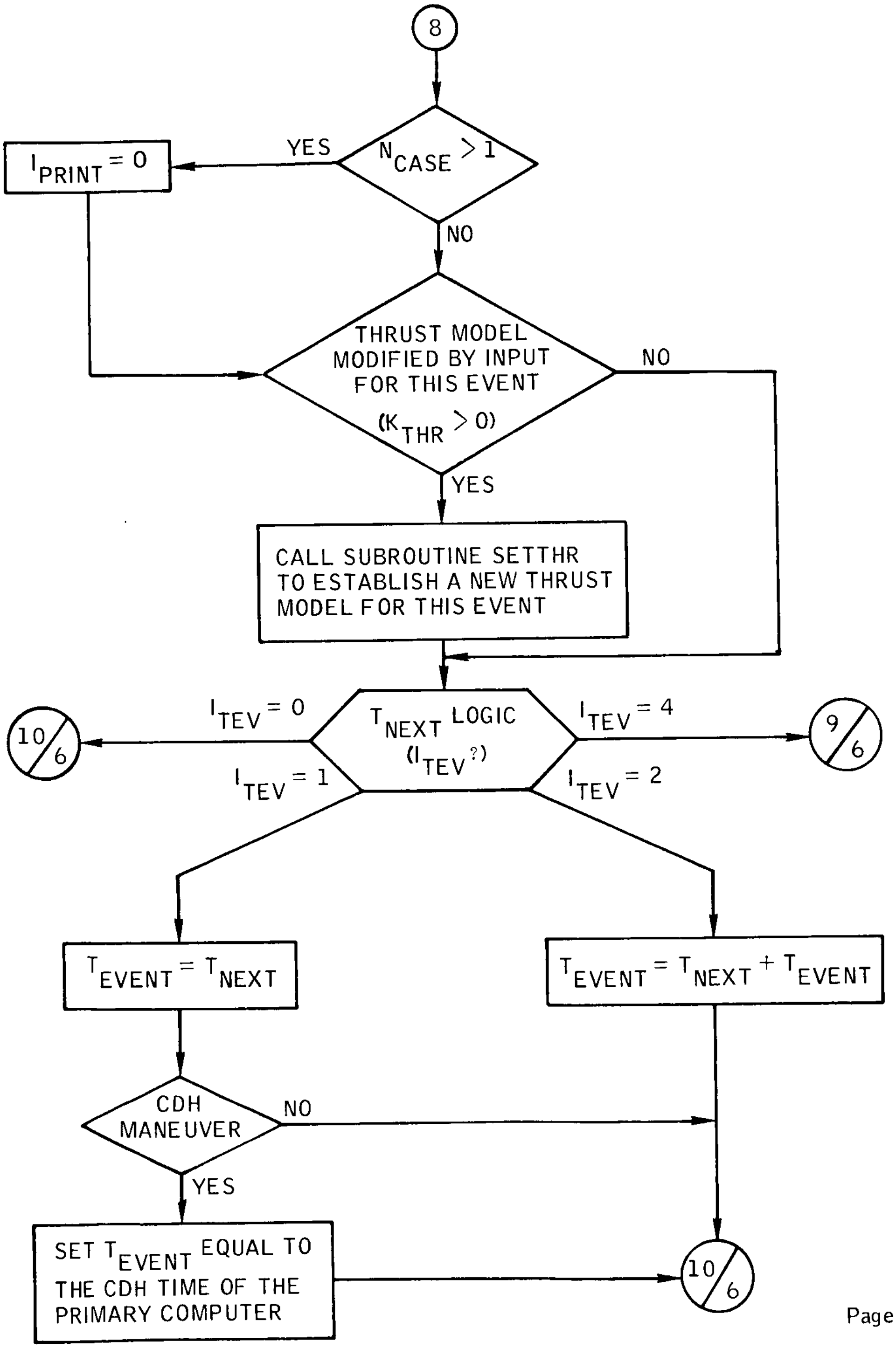

8
$N_{CASE} > 1$
YES
NO
$I_{PRINT} = 0$
THRUST MODEL MODIFIED BY INPUT FOR THIS EVENT ($K_{THR} > 0$)
NO
YES
CALL SUBROUTINE SETTHR TO ESTABLISH A NEW THRUST MODEL FOR THIS EVENT
T_{NEXT} LOGIC (I_{TEV}?)
$I_{TEV} = 0$
$I_{TEV} = 4$
$I_{TEV} = 1$
$I_{TEV} = 2$
10/6
9/6
$T_{EVENT} = T_{NEXT}$
$T_{EVENT} = T_{NEXT} + T_{EVENT}$
CDH MANEUVER
NO
YES
SET T_{EVENT} EQUAL TO THE CDH TIME OF THE PRIMARY COMPUTER
10/6

Figure B-8.- Continued.

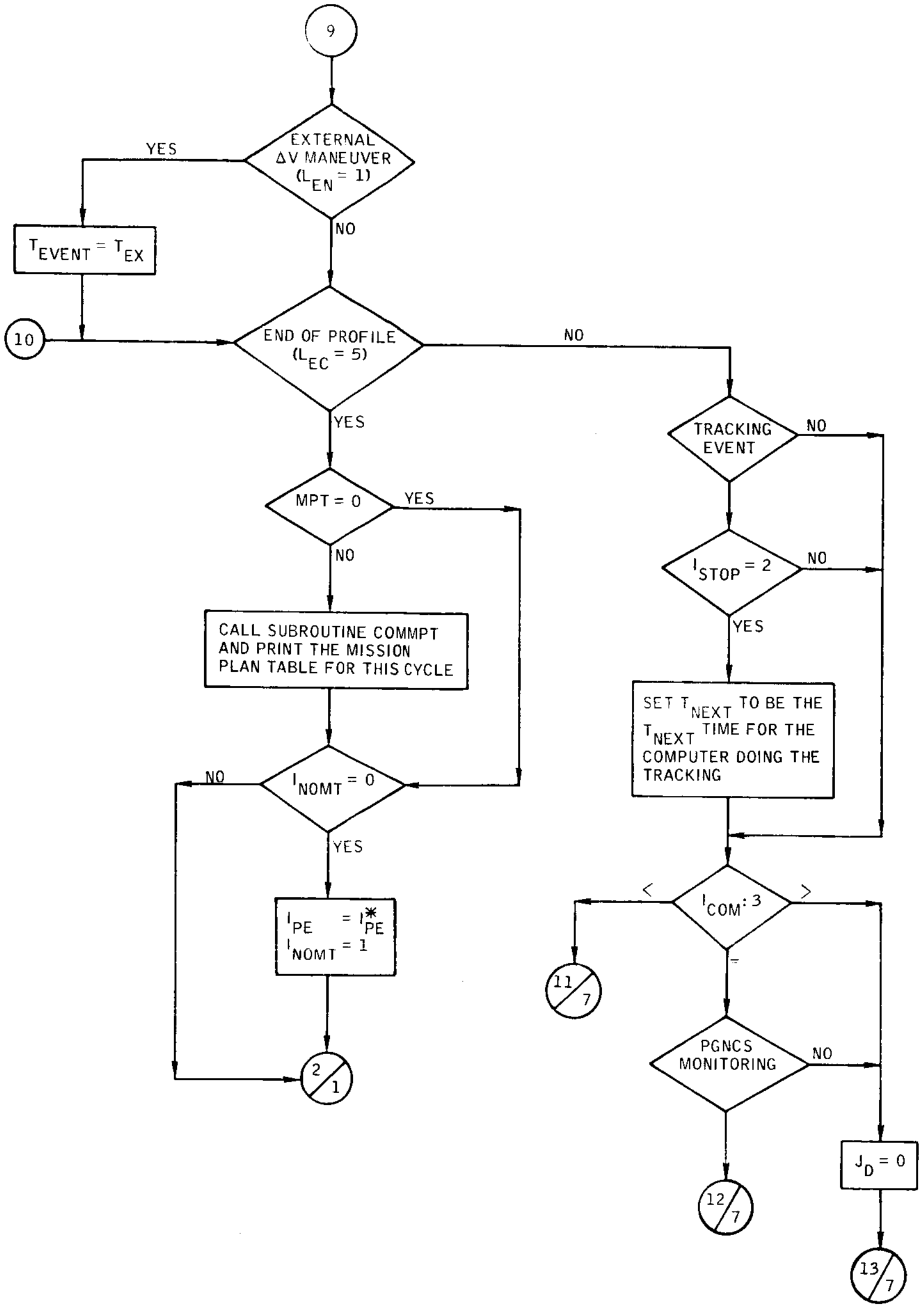
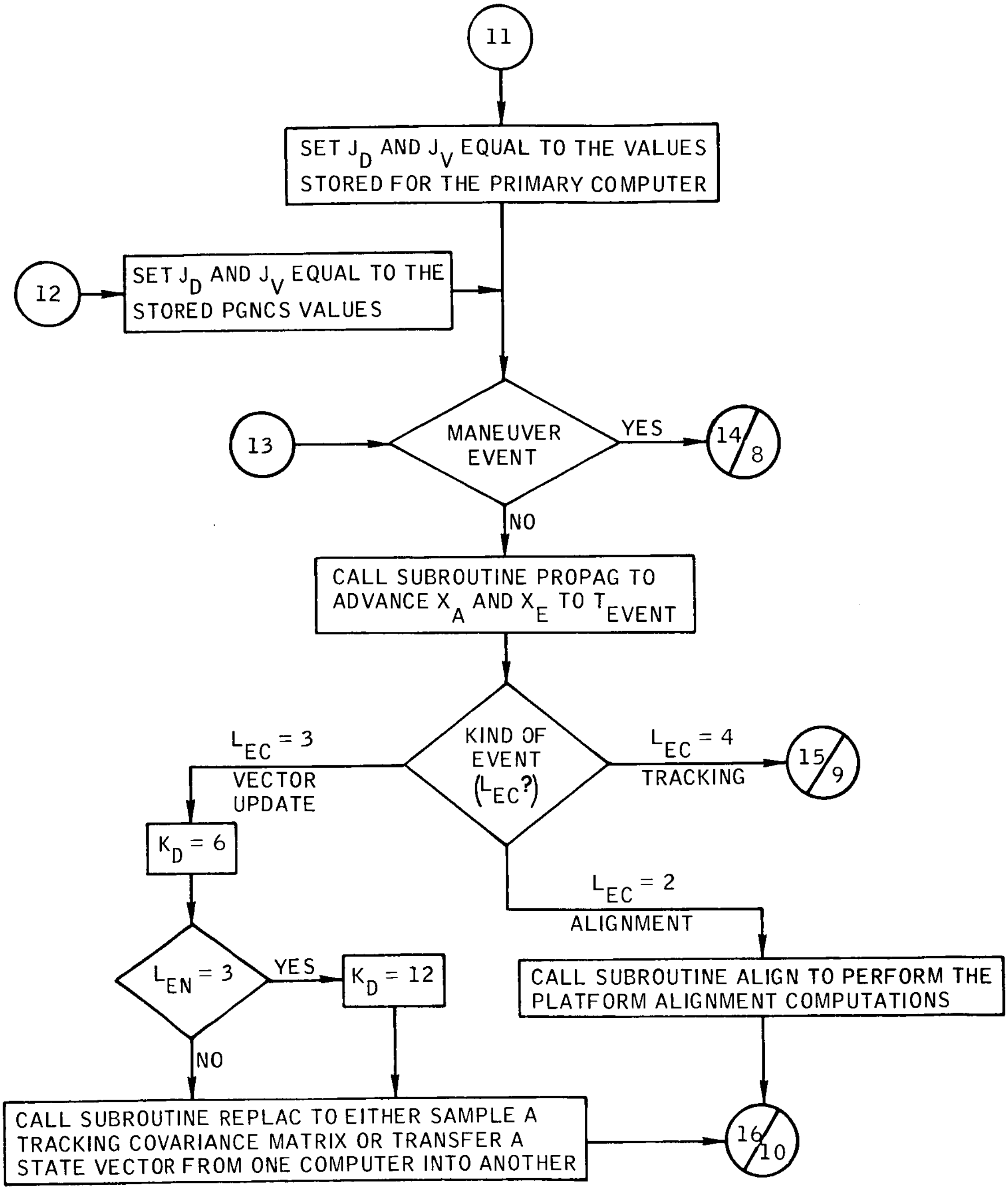

9
EXTERNAL ΔV MANEUVER (L_{EN} = 1)
YES
NO
$T_{EVENT} = T_{EX}$
10
END OF PROFILE (L_{EC} = 5)
NO
YES
MPT = 0
YES
NO
CALL SUBROUTINE COMMPT AND PRINT THE MISSION PLAN TABLE FOR THIS CYCLE
I_{NOMT} = 0
NO
YES
$I_{PE} = I^*_{PE}$
I_{NOMT} = 1
2/1
TRACKING EVENT
NO
I_{STOP} = 2
NO
YES
SET T_{NEXT} TO BE THE T_{NEXT} TIME FOR THE COMPUTER DOING THE TRACKING
I_{COM} : 3
<
>
=
11/7
PGNCS MONITORING
NO
12/7
J_D = 0
13/7

Figure B-8.- Continued.


11
SET J_D AND J_V EQUAL TO THE VALUES STORED FOR THE PRIMARY COMPUTER
12
SET J_D AND J_V EQUAL TO THE STORED PGNCS VALUES
13
MANEUVER EVENT
YES
14/8
NO
CALL SUBROUTINE PROPAG TO ADVANCE X_A AND X_E TO T_{EVENT}
KIND OF EVENT (L_{EC}?)
$L_{EC} = 3$
VECTOR UPDATE
$L_{EC} = 4$
TRACKING
15/9
$K_D = 6$
$L_{EC} = 2$
ALIGNMENT
$L_{EN} = 3$
YES
$K_D = 12$
CALL SUBROUTINE ALIGN TO PERFORM THE PLATFORM ALIGNMENT COMPUTATIONS
NO
CALL SUBROUTINE REPLAC TO EITHER SAMPLE A TRACKING COVARIANCE MATRIX OR TRANSFER A STATE VECTOR FROM ONE COMPUTER INTO ANOTHER
16/10

Page 7 of 10

Figure B-8.- Continued.

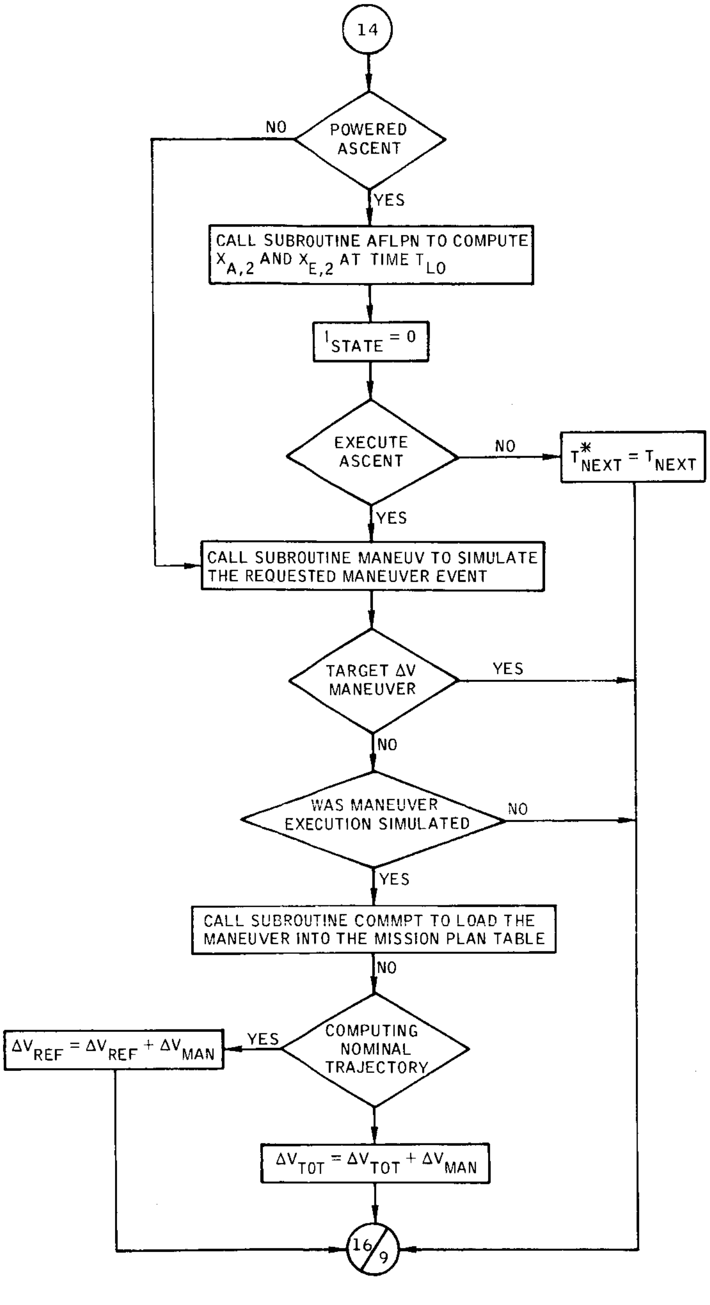
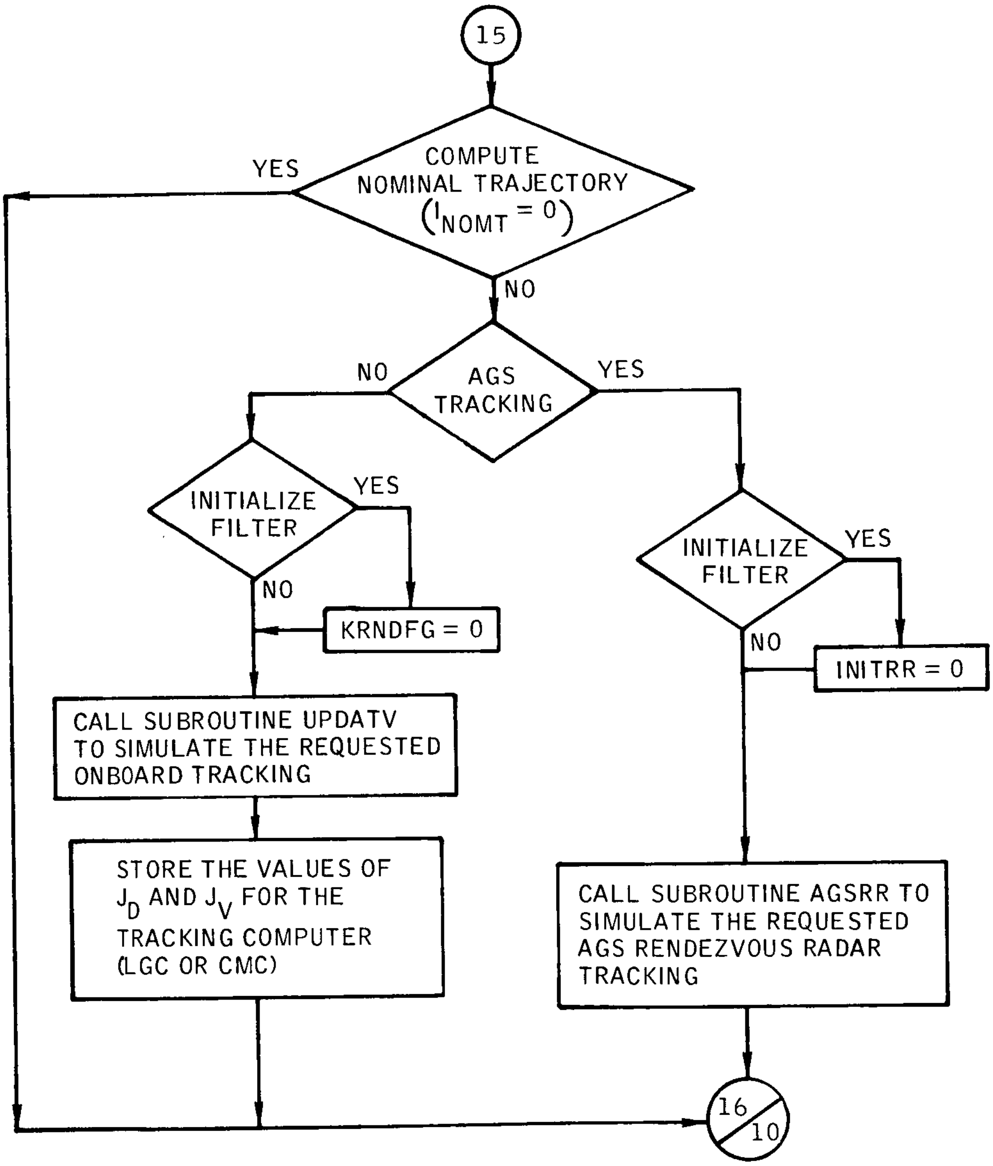

14
POWERED ASCENT
NO
YES
CALL SUBROUTINE AFLPN TO COMPUTE $X_{A,2}$ AND $X_{E,2}$ AT TIME T_{LO}
$I_{STATE} = 0$
EXECUTE ASCENT
NO
$T^*_{NEXT} = T_{NEXT}$
YES
CALL SUBROUTINE MANEUV TO SIMULATE THE REQUESTED MANEUVER EVENT
TARGET ΔV MANEUVER
YES
NO
WAS MANEUVER EXECUTION SIMULATED
NO
YES
CALL SUBROUTINE COMMPT TO LOAD THE MANEUVER INTO THE MISSION PLAN TABLE
NO
COMPUTING NOMINAL TRAJECTORY
YES
$\Delta V_{REF} = \Delta V_{REF} + \Delta V_{MAN}$
$\Delta V_{TOT} = \Delta V_{TOT} + \Delta V_{MAN}$
16/9

Figure B-8.- Continued.


15
COMPUTE NOMINAL TRAJECTORY ($I_{NOMT} = 0$)
YES
NO
AGS TRACKING
NO
YES
INITIALIZE FILTER
YES
NO
KRNDFG = 0
CALL SUBROUTINE UPDATV TO SIMULATE THE REQUESTED ONBOARD TRACKING
STORE THE VALUES OF J_D AND J_V FOR THE TRACKING COMPUTER (LGC OR CMC)
INITIALIZE FILTER
YES
NO
INITRR = 0
CALL SUBROUTINE AGSRR TO SIMULATE THE REQUESTED AGS RENDEZVOUS RADAR TRACKING
16
10

Page 9 of 10

Figure B-8.- Continued.

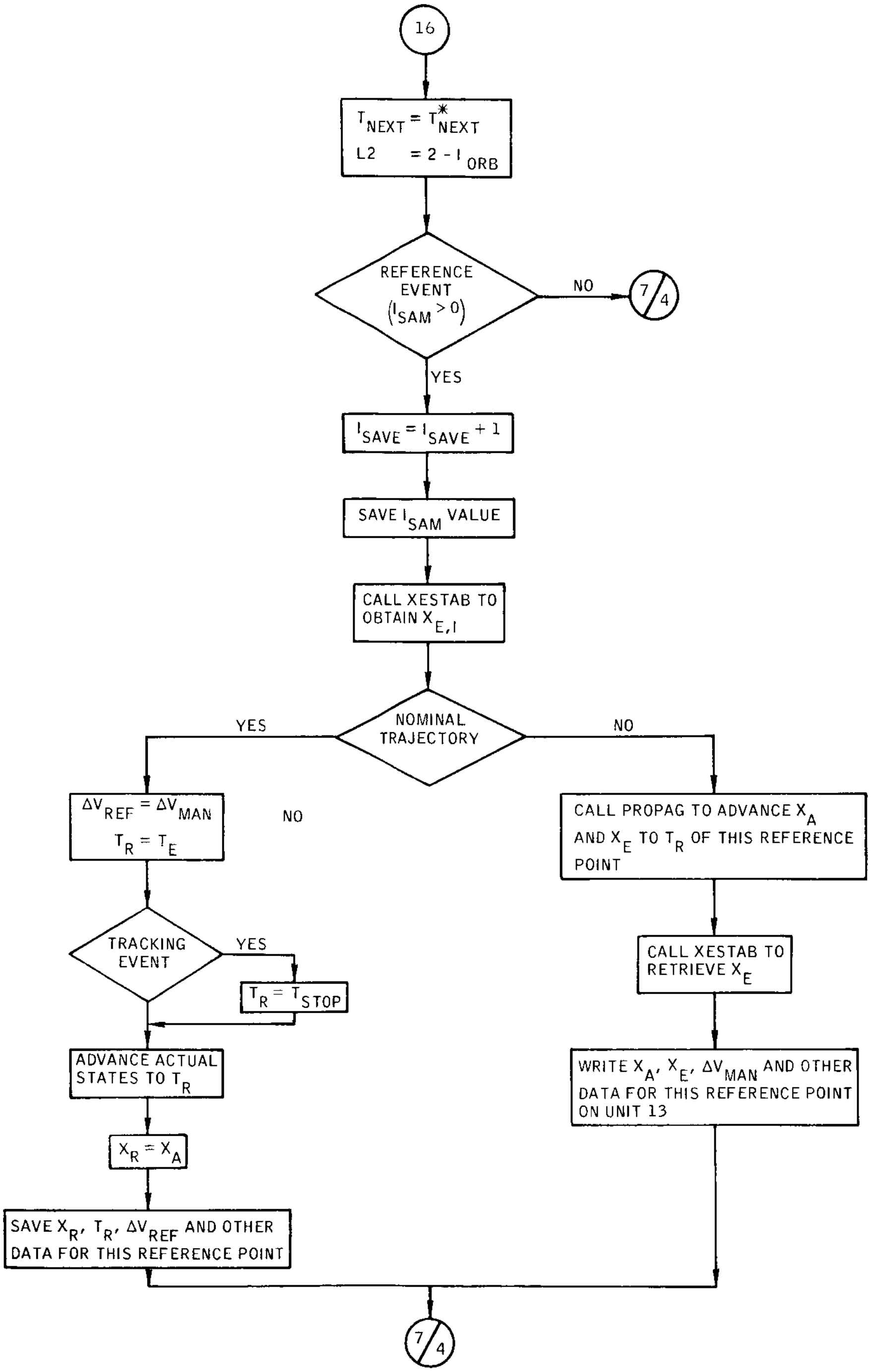
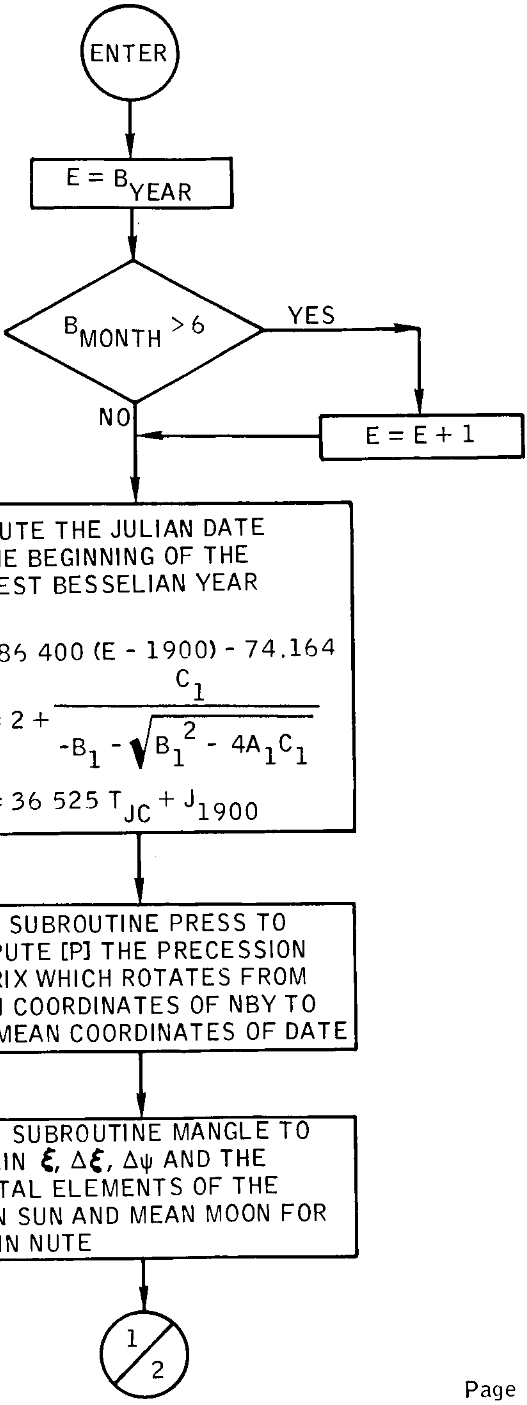
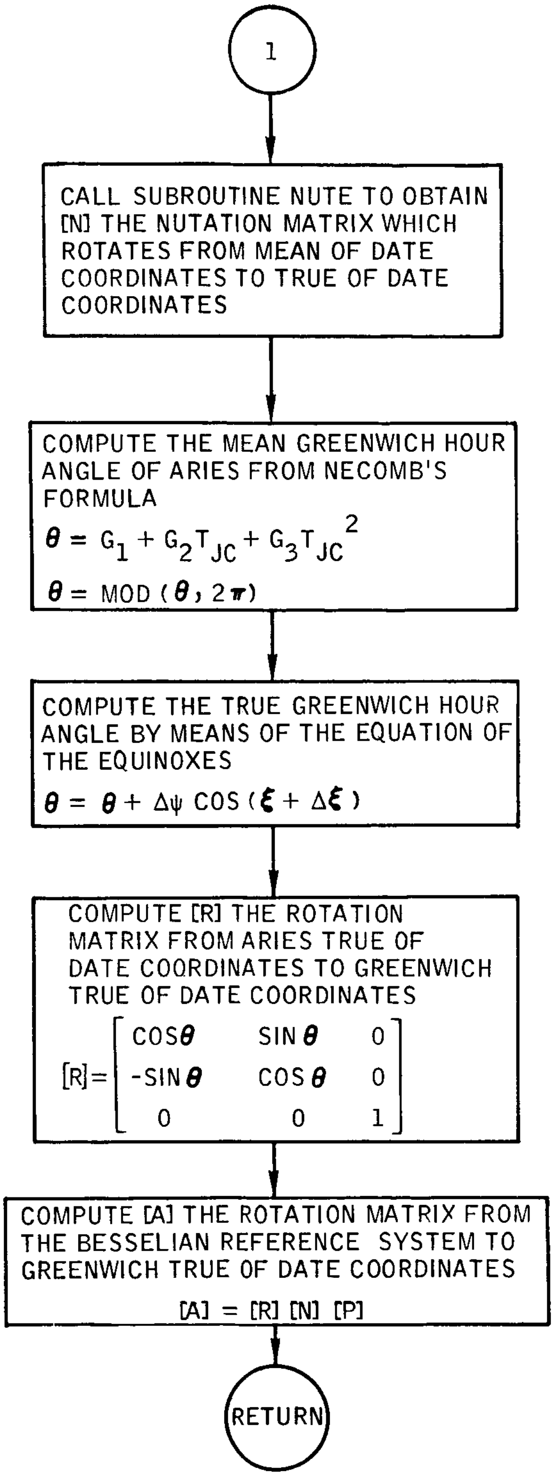

16
$T_{NEXT} = T^*_{NEXT}$
$L2 = 2 - I_{ORB}$
REFERENCE EVENT ($I_{SAM} > 0$)
NO
7/4
YES
$I_{SAVE} = I_{SAVE} + 1$
SAVE I_{SAM} VALUE
CALL XESTAB TO OBTAIN $X_{E,I}$
NOMINAL TRAJECTORY
YES
NO
$\Delta V_{REF} = \Delta V_{MAN}$
$T_R = T_E$
NO
CALL PROPAG TO ADVANCE X_A AND X_E TO T_R OF THIS REFERENCE POINT
TRACKING EVENT
YES
$T_R = T_{STOP}$
CALL XESTAB TO RETRIEVE X_E
ADVANCE ACTUAL STATES TO T_R
WRITE X_A, X_E, ΔV_{MAN} AND OTHER DATA FOR THIS REFERENCE POINT ON UNIT 13
$X_R = X_A$
SAVE X_R, T_R, ΔV_{REF} AND OTHER DATA FOR THIS REFERENCE POINT
7/4

Figure B-8.- Concluded.


ENTER
$E = B_{YEAR}$
$B_{MONTH} > 6$
YES
NO
$E = E + 1$
COMPUTE THE JULIAN DATE OF THE BEGINNING OF THE NEAREST BESSELIAN YEAR (NBY)
$C_1 = 86\,400\,(E - 1900) - 74.164$
$T_{JC} = 2 + \frac{C_1}{-B_1 - \sqrt{B_1^2 - 4A_1C_1}}$
$T_{JD} = 36\,525\,T_{JC} + J_{1900}$
CALL SUBROUTINE PRESS TO COMPUTE [P] THE PRECESSION MATRIX WHICH ROTATES FROM MEAN COORDINATES OF NBY TO THE MEAN COORDINATES OF DATE
CALL SUBROUTINE MANGLE TO OBTAIN ϵ, $\Delta\epsilon$, $\Delta\psi$ AND THE ORBITAL ELEMENTS OF THE MEAN SUN AND MEAN MOON FOR USE IN NUTE
1
2

Page 1 of 2

Figure B-9.- Flow chart of subroutine AMATRX.



CALL SUBROUTINE NUTE TO OBTAIN [N] THE NUTATION MATRIX WHICH ROTATES FROM MEAN OF DATE COORDINATES TO TRUE OF DATE COORDINATES

COMPUTE THE MEAN GREENWICH HOUR ANGLE OF ARIES FROM NECOMB'S FORMULA

$$\theta = G_1 + G_2 T_{JC} + G_3 T_{JC}^{\ 2}$$

$$\theta = \mathrm{MOD}\,(\theta, 2\pi)$$

COMPUTE THE TRUE GREENWICH HOUR ANGLE BY MEANS OF THE EQUATION OF THE EQUINOXES

$$\theta = \theta + \Delta\psi \,\mathrm{COS}\,(\xi + \Delta\xi)$$

COMPUTE [R] THE ROTATION MATRIX FROM ARIES TRUE OF DATE COORDINATES TO GREENWICH TRUE OF DATE COORDINATES

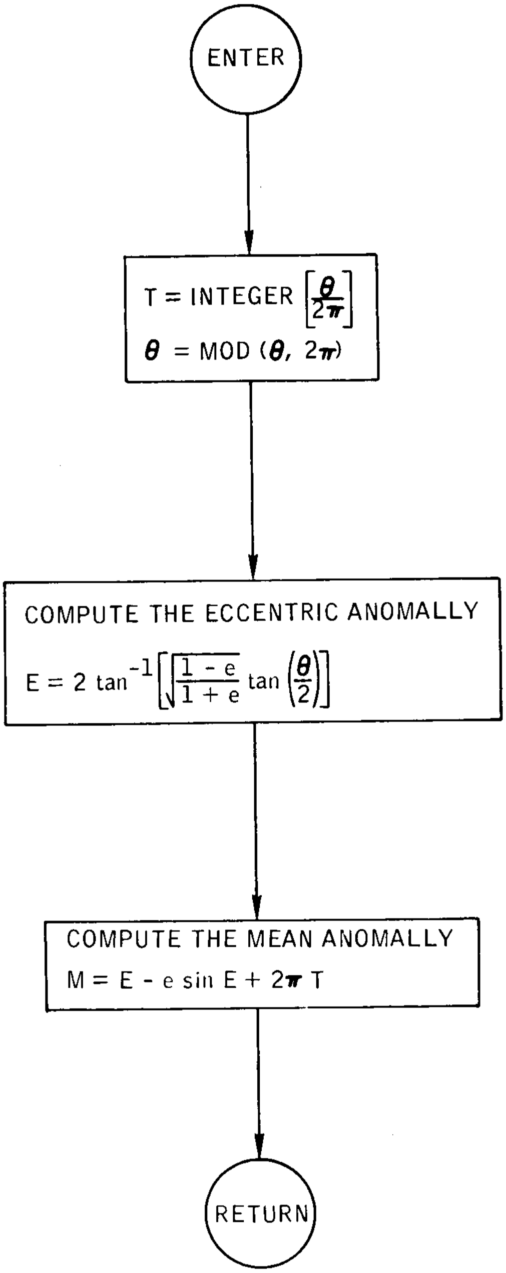
$$[R] = \begin{bmatrix} \mathrm{COS}\,\theta & \mathrm{SIN}\,\theta & 0 \\ -\mathrm{SIN}\,\theta & \mathrm{COS}\,\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

COMPUTE [A] THE ROTATION MATRIX FROM THE BESSELIAN REFERENCE SYSTEM TO GREENWICH TRUE OF DATE COORDINATES

$$[A] = [R]\,[N]\,[P]$$

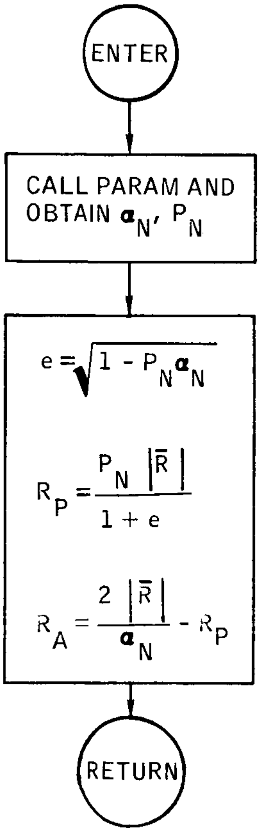
Page 2 of 2

Figure B-9.- Concluded.


ENTER
T = INTEGER $\left[\frac{\theta}{2\pi}\right]$
θ = MOD (θ, 2π)
COMPUTE THE ECCENTRIC ANOMALLY
$E = 2\tan^{-1}\left[\sqrt{\frac{1-e}{1+e}}\tan\left(\frac{\theta}{2}\right)\right]$
COMPUTE THE MEAN ANOMALLY
$M = E - e\sin E + 2\pi T$
RETURN

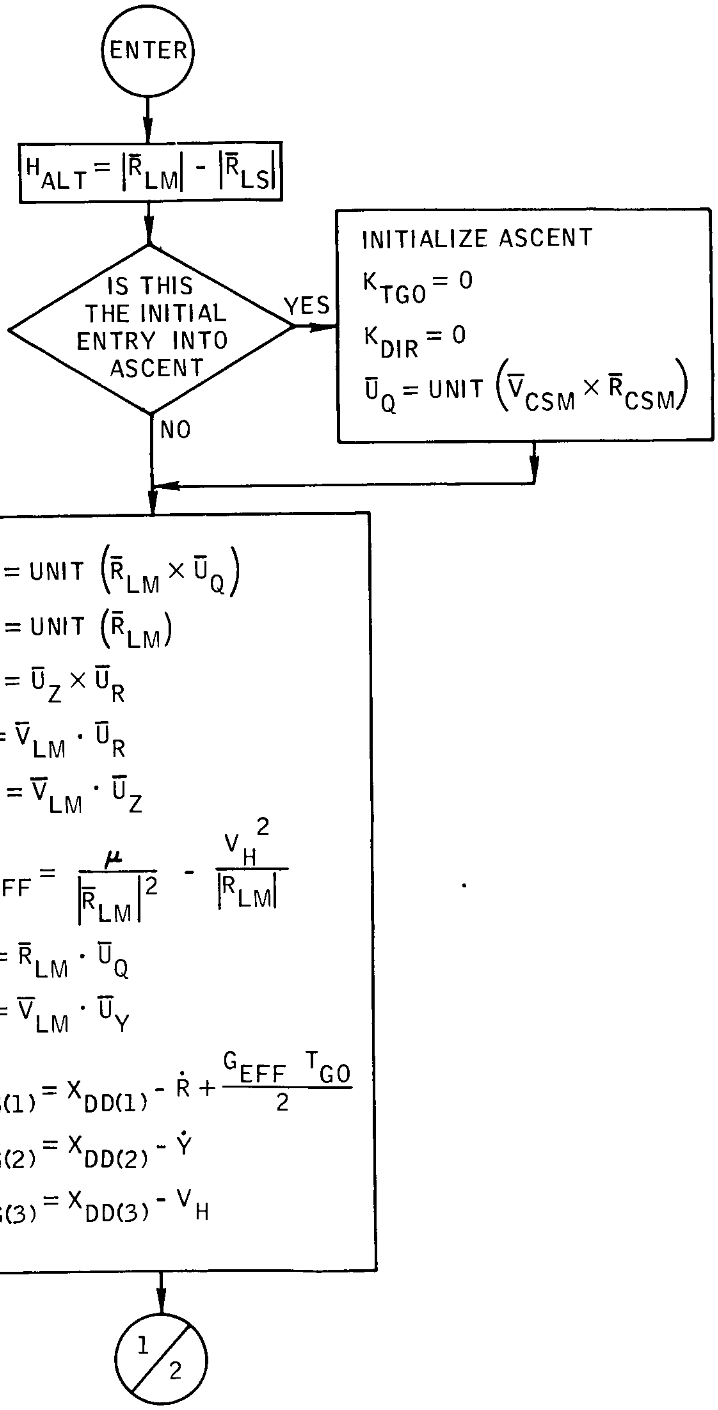
Page 1 of 1

Figure B-10.- Flow chart of subroutine ANOM.


ENTER
CALL PARAM AND OBTAIN α_N, P_N
$e = \sqrt{1 - P_N \alpha_N}$
$R_P = \frac{P_N |\bar{R}|}{1 + e}$
$R_A = \frac{2 |\bar{R}|}{\alpha_N} - R_P$
RETURN

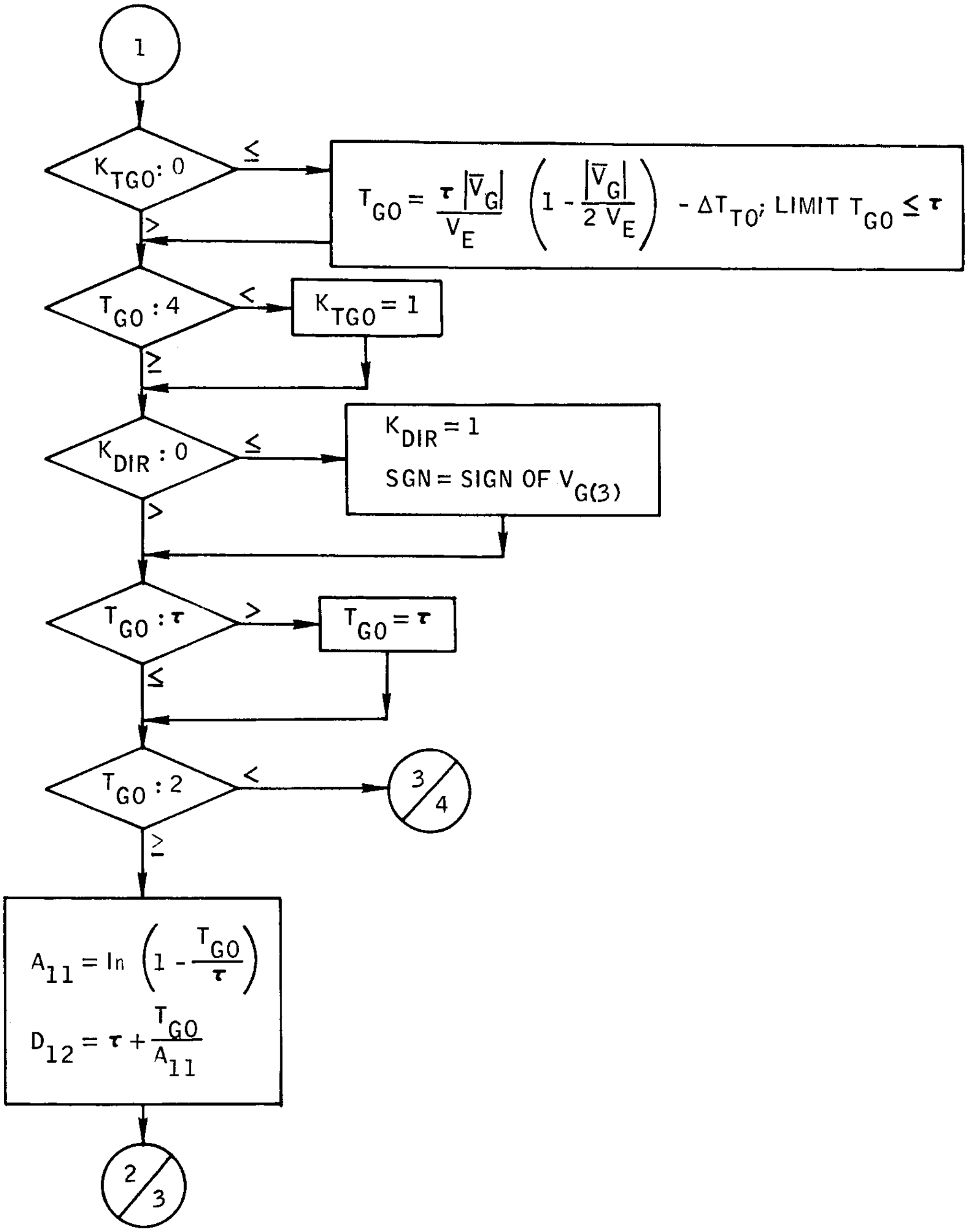
Page 1 of 1

Figure B-11.- Flow chart of subroutine APSIDE.


ENTER
$H_{ALT} = |\bar{R}_{LM}| - |\bar{R}_{LS}|$
IS THIS THE INITIAL ENTRY INTO ASCENT
YES
NO
INITIALIZE ASCENT
$K_{TGO} = 0$
$K_{DIR} = 0$
$\bar{U}_Q = \text{UNIT}\left(\bar{V}_{CSM} \times \bar{R}_{CSM}\right)$
$\bar{U}_Z = \text{UNIT}\left(\bar{R}_{LM} \times \bar{U}_Q\right)$
$\bar{U}_R = \text{UNIT}\left(\bar{R}_{LM}\right)$
$\bar{U}_Y = \bar{U}_Z \times \bar{U}_R$
$\dot{R} = \bar{V}_{LM} \cdot \bar{U}_R$
$V_H = \bar{V}_{LM} \cdot \bar{U}_Z$
$G_{EFF} = \frac{\mu}{|\bar{R}_{LM}|^2} - \frac{V_H^2}{|\bar{R}_{LM}|}$
$Y = \bar{R}_{LM} \cdot \bar{U}_Q$
$\dot{Y} = \bar{V}_{LM} \cdot \bar{U}_Y$
$V_{G(1)} = X_{DD(1)} - \dot{R} + \frac{G_{EFF} \; T_{GO}}{2}$
$V_{G(2)} = X_{DD(2)} - \dot{Y}$
$V_{G(3)} = X_{DD(3)} - V_H$
1/2

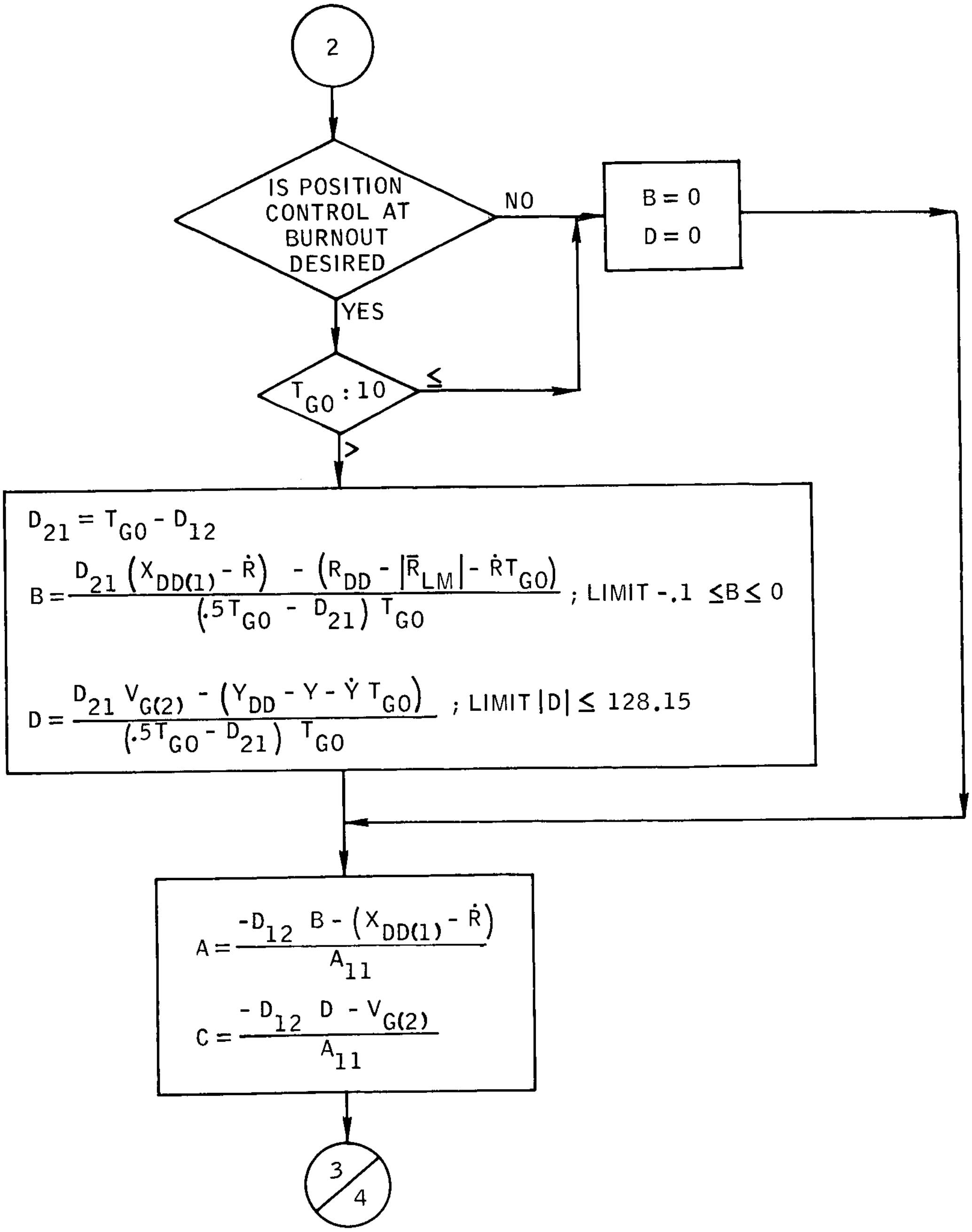
Page 1 of 5

Figure B-12.- Flow chart of subroutine ASCENT.



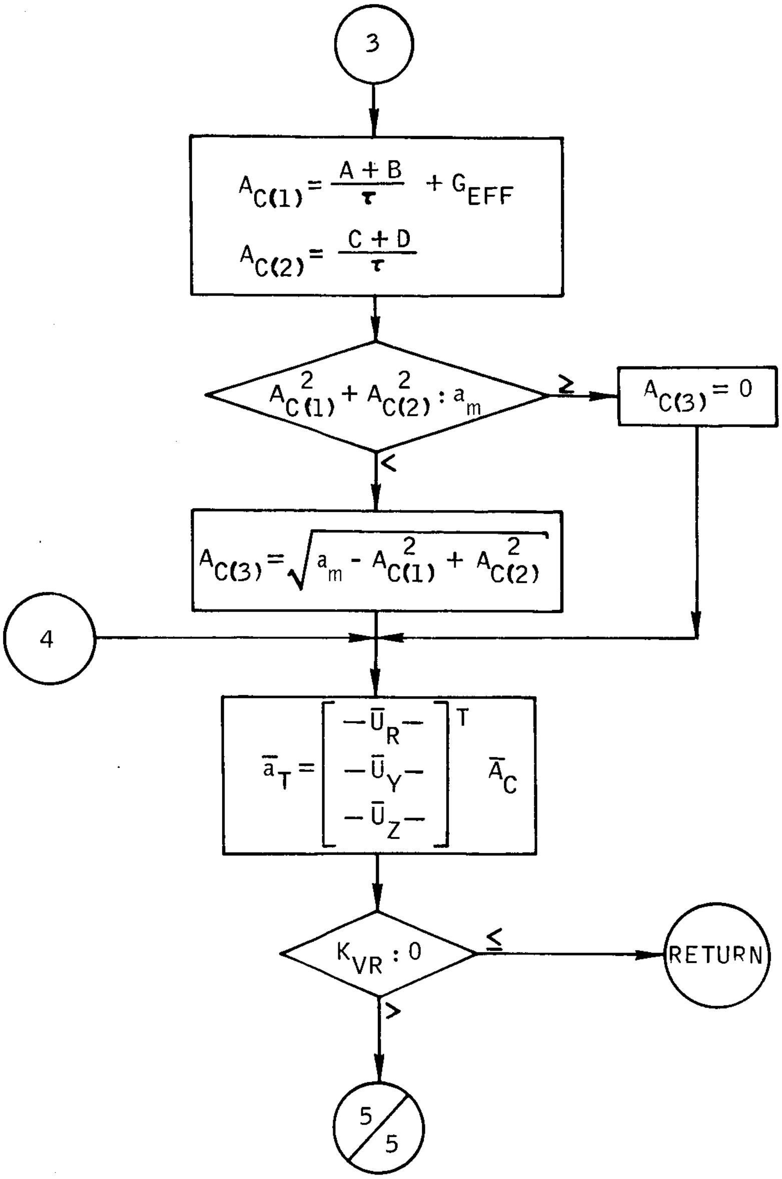
Page 2 of 5

Figure B-12.- Continued.


2
IS POSITION CONTROL AT BURNOUT DESIRED
NO
YES
$B = 0$
$D = 0$
$T_{GO} : 10$
$\leq$
$>$
$D_{21} = T_{GO} - D_{12}$
$B = \frac{D_{21}\left(X_{DD(1)} - \dot{R}\right) - \left(R_{DD} - |\bar{R}_{LM}| - \dot{R}T_{GO}\right)}{\left(.5T_{GO} - D_{21}\right) T_{GO}}$; LIMIT $-.1 \leq B \leq 0$
$D = \frac{D_{21} V_{G(2)} - \left(Y_{DD} - Y - \dot{Y} T_{GO}\right)}{\left(.5T_{GO} - D_{21}\right) T_{GO}}$; LIMIT $|D| \leq 128.15$
$A = \frac{-D_{12} B - \left(X_{DD(1)} - \dot{R}\right)}{A_{11}}$
$C = \frac{-D_{12} D - V_{G(2)}}{A_{11}}$
3
4

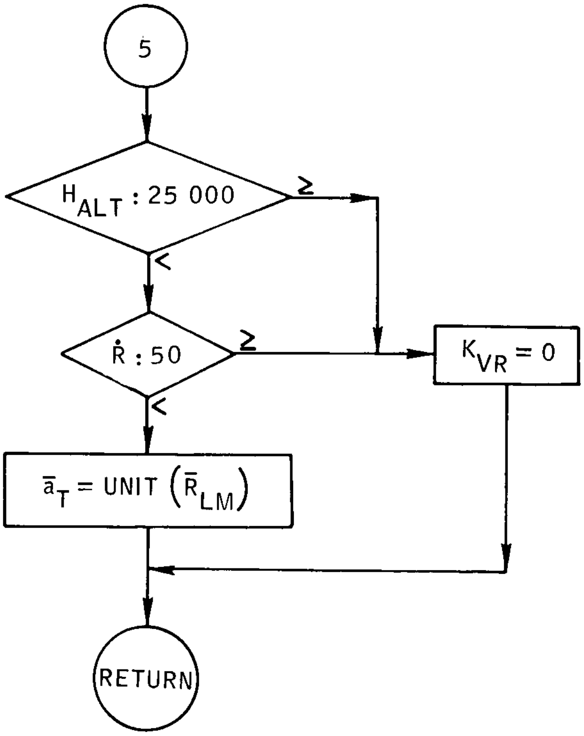
Page 3 of 5

Figure B-12.- Continued.


3
$A_{C(1)} = \frac{A+B}{\tau} + G_{EFF}$
$A_{C(2)} = \frac{C+D}{\tau}$
$A_{C(1)}^2 + A_{C(2)}^2 : a_m$
≥
$A_{C(3)} = 0$
<
$A_{C(3)} = \sqrt{a_m - A_{C(1)}^2 + A_{C(2)}^2}$
4
$\bar{a}_T = \begin{bmatrix} -\bar{U}_R- \\ -\bar{U}_Y- \\ -\bar{U}_Z- \end{bmatrix}^T \bar{A}_C$
$K_{VR} : 0$
≤
RETURN
>
5/5

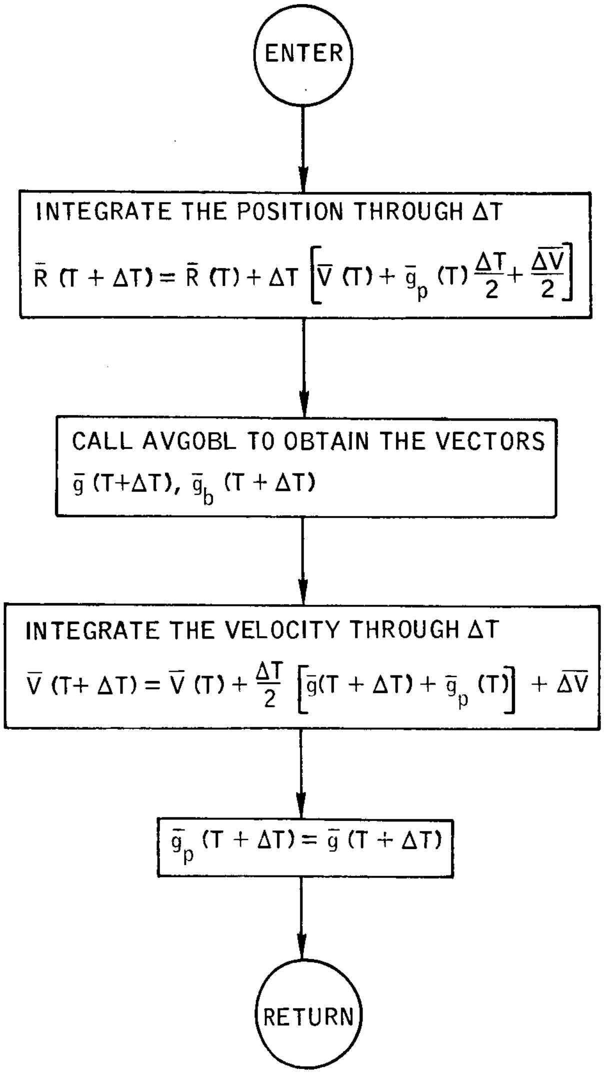
Page 4 of 5

Figure B-12.- Continued.



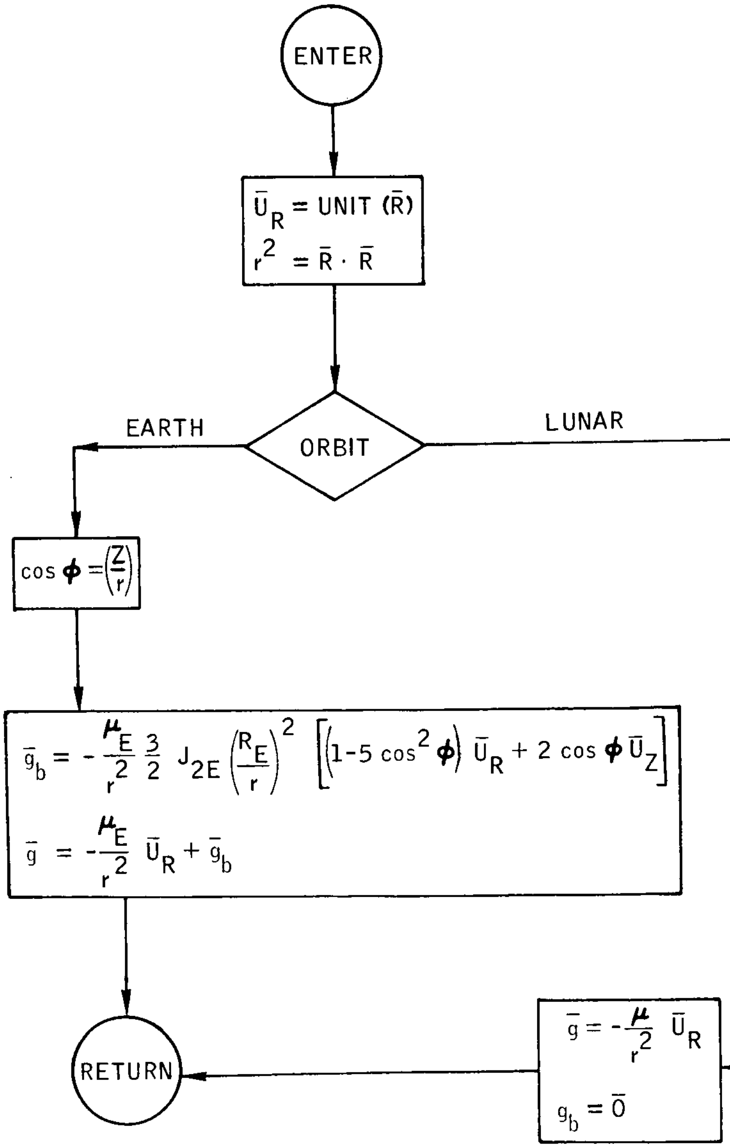
Page 5 of 5

Figure B-12.- Concluded.


ENTER
INTEGRATE THE POSITION THROUGH ΔT
$\bar{R}(T+\Delta T)=\bar{R}(T)+\Delta T\left[\bar{V}(T)+\bar{g}_p(T)\frac{\Delta T}{2}+\frac{\overline{\Delta V}}{2}\right]$
CALL AVGOBL TO OBTAIN THE VECTORS
$\bar{g}(T+\Delta T)$, $\bar{g}_b(T+\Delta T)$
INTEGRATE THE VELOCITY THROUGH ΔT
$\bar{V}(T+\Delta T)=\bar{V}(T)+\frac{\Delta T}{2}\left[\bar{g}(T+\Delta T)+\bar{g}_p(T)\right]+\overline{\Delta V}$
$\bar{g}_p(T+\Delta T)=\bar{g}(T+\Delta T)$
RETURN

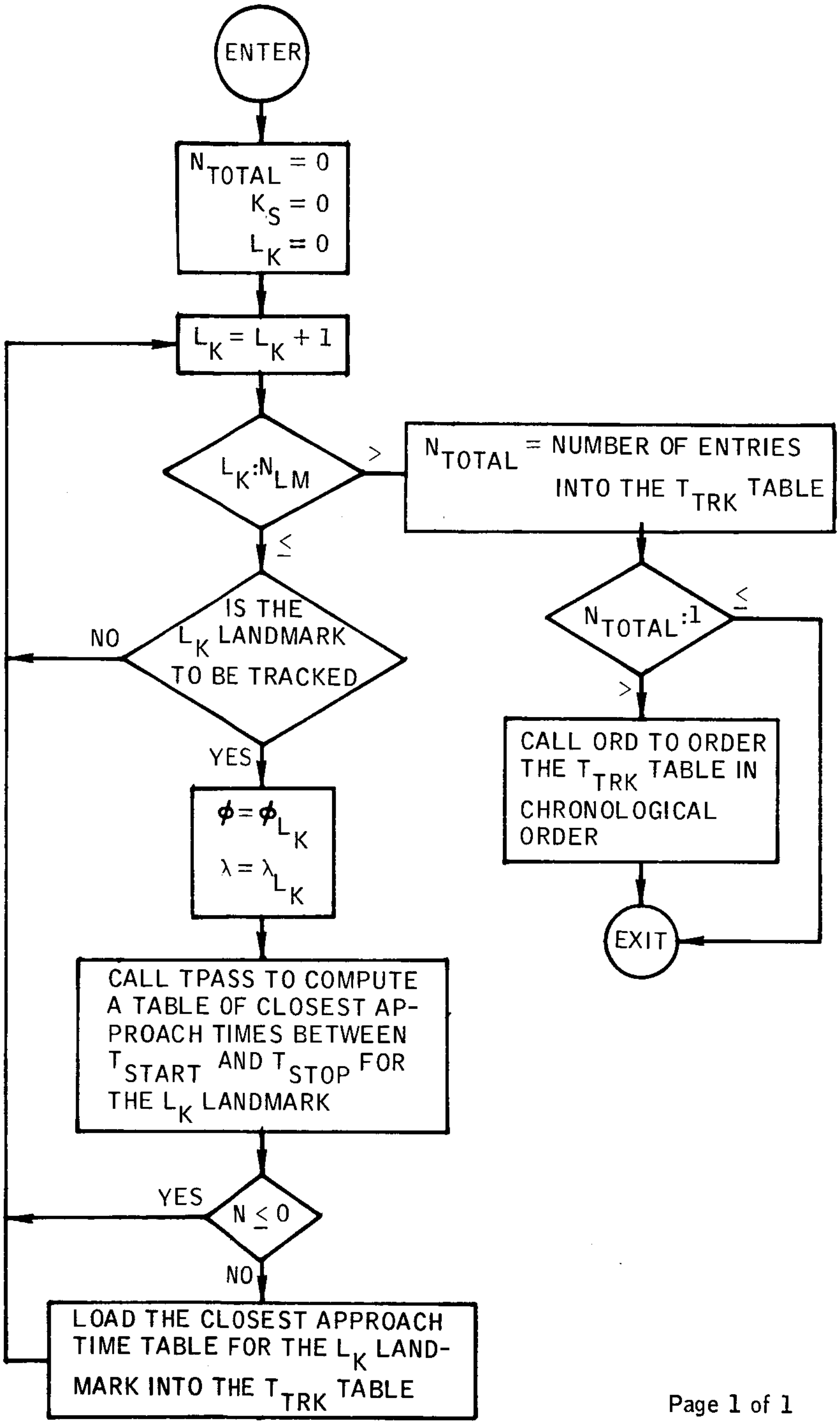
Page 1 of 1

Figure B-13.- Flow chart of subroutine AVEG.


ENTER
$\bar{U}_R = \text{UNIT}(\bar{R})$
$r^2 = \bar{R} \cdot \bar{R}$
ORBIT
EARTH
LUNAR
$\cos \phi = \left(\frac{Z}{r}\right)$
$\bar{g}_b = -\frac{\mu_E}{r^2} \frac{3}{2} J_{2E} \left(\frac{R_E}{r}\right)^2 \left[\left(1 - 5 \cos^2 \phi\right) \bar{U}_R + 2 \cos \phi \bar{U}_Z\right]$
$\bar{g} = -\frac{\mu_E}{r^2} \bar{U}_R + \bar{g}_b$
$\bar{g} = -\frac{\mu}{r^2} \bar{U}_R$
$g_b = \bar{0}$
RETURN

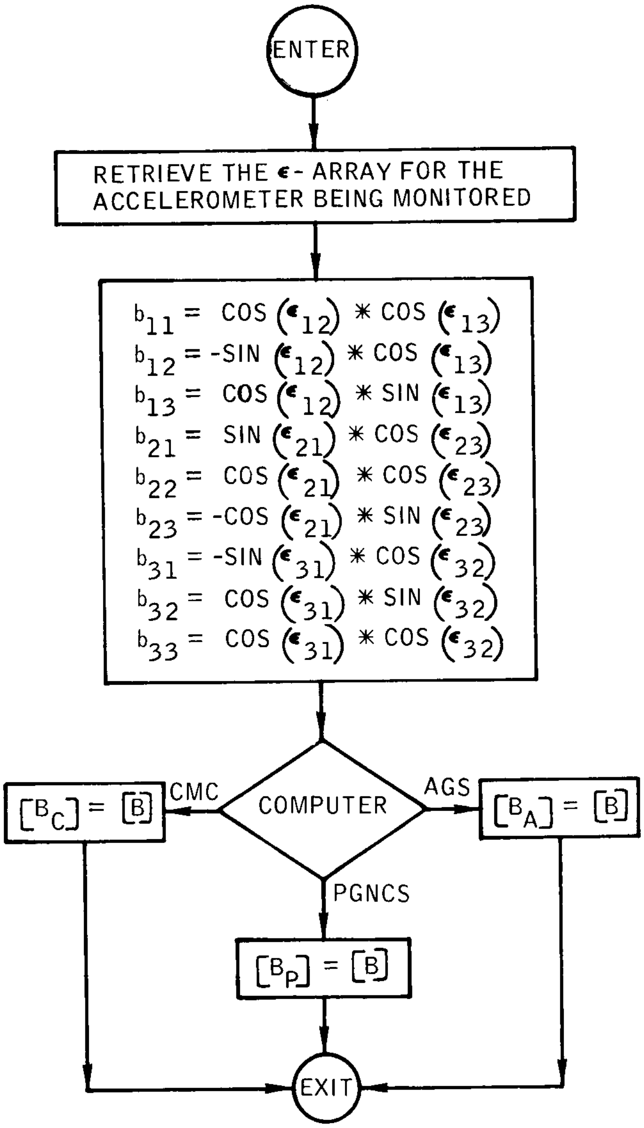
Page 1 of 1

Figure B-14.- Flow chart of subroutine AVGOBL.


ENTER
$N_{TOTAL} = 0$
$K_S = 0$
$L_K = 0$
$L_K = L_K + 1$
$L_K:N_{LM}$
>
N_{TOTAL} = NUMBER OF ENTRIES INTO THE T_{TRK} TABLE
≤
IS THE L_K LANDMARK TO BE TRACKED
NO
YES
$\phi = \phi_{L_K}$
$\lambda = \lambda_{L_K}$
CALL TPASS TO COMPUTE A TABLE OF CLOSEST APPROACH TIMES BETWEEN T_{START} AND T_{STOP} FOR THE L_K LANDMARK
$N \leq 0$
YES
NO
LOAD THE CLOSEST APPROACH TIME TABLE FOR THE L_K LANDMARK INTO THE T_{TRK} TABLE
$N_{TOTAL}:1$
≤
>
CALL ORD TO ORDER THE T_{TRK} TABLE IN CHRONOLOGICAL ORDER
EXIT

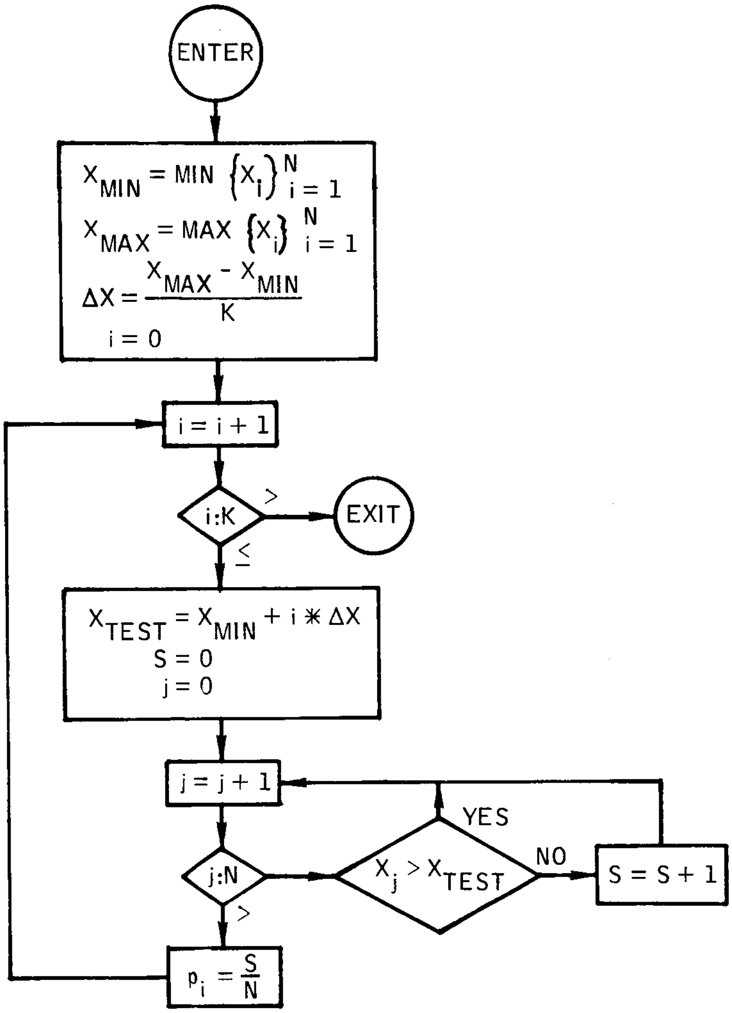
Page 1 of 1

Figure B-15.- Flow chart of subroutine CATIME.


ENTER
RETRIEVE THE ϵ- ARRAY FOR THE ACCELEROMETER BEING MONITORED
b11 = COS (ϵ12) ✱ COS (ϵ13)
b12 = -SIN (ϵ12) ✱ COS (ϵ13)
b13 = COS (ϵ12) ✱ SIN (ϵ13)
b21 = SIN (ϵ21) ✱ COS (ϵ23)
b22 = COS (ϵ21) ✱ COS (ϵ23)
b23 = -COS (ϵ21) ✱ SIN (ϵ23)
b31 = -SIN (ϵ31) ✱ COS (ϵ32)
b32 = COS (ϵ31) ✱ SIN (ϵ32)
b33 = COS (ϵ31) ✱ COS (ϵ32)
COMPUTER
CMC
AGS
PGNCS
[BC] = [B]
[BA] = [B]
[BP] = [B]
EXIT

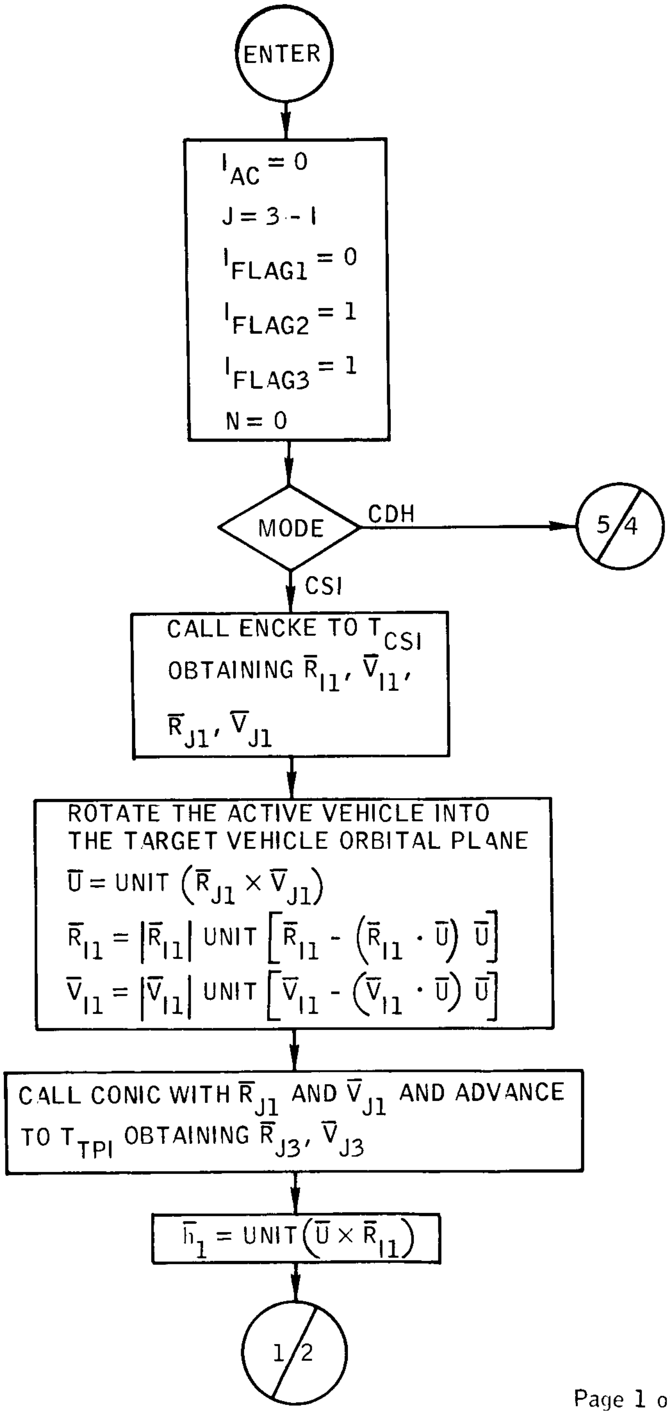
Page 1 of 1

Figure B-16.- Flow chart of subroutine CBA.


ENTER
$X_{MIN} = MIN \{X_i\}_{i=1}^{N}$
$X_{MAX} = MAX \{X_i\}_{i=1}^{N}$
$\Delta X = \frac{X_{MAX} - X_{MIN}}{K}$
$i = 0$
$i = i + 1$
i:K
>
EXIT
≤
$X_{TEST} = X_{MIN} + i * \Delta X$
$S = 0$
$j = 0$
$j = j + 1$
j:N
$X_j > X_{TEST}$
YES
NO
$S = S + 1$
>
$p_i = \frac{S}{N}$

Page 1 of 1

Figure B-17.- Flow chart of subroutine CDFUNC.


ENTER
$I_{AC} = 0$
$J = 3 - I$
$I_{FLAG1} = 0$
$I_{FLAG2} = 1$
$I_{FLAG3} = 1$
$N = 0$
MODE
CDH
5/4
CSI
CALL ENCKE TO T_{CSI} OBTAINING $\bar{R}_{I1}$, $\bar{V}_{I1}$, $\bar{R}_{J1}$, $\bar{V}_{J1}$
ROTATE THE ACTIVE VEHICLE INTO THE TARGET VEHICLE ORBITAL PLANE
$\bar{U} = \text{UNIT}\left(\bar{R}_{J1} \times \bar{V}_{J1}\right)$
$\bar{R}_{I1} = \left|\bar{R}_{I1}\right| \text{UNIT}\left[\bar{R}_{I1} - \left(\bar{R}_{I1} \cdot \bar{U}\right) \bar{U}\right]$
$\bar{V}_{I1} = \left|\bar{V}_{I1}\right| \text{UNIT}\left[\bar{V}_{I1} - \left(\bar{V}_{I1} \cdot \bar{U}\right) \bar{U}\right]$
CALL CONIC WITH $\bar{R}_{J1}$ AND $\bar{V}_{J1}$ AND ADVANCE TO T_{TPI} OBTAINING $\bar{R}_{J3}$, $\bar{V}_{J3}$
$\bar{h}_1 = \text{UNIT}\left(\bar{U} \times \bar{R}_{I1}\right)$
1/2

Page 1 of 9

Figure B-18.- Flow chart of subroutine CFP.

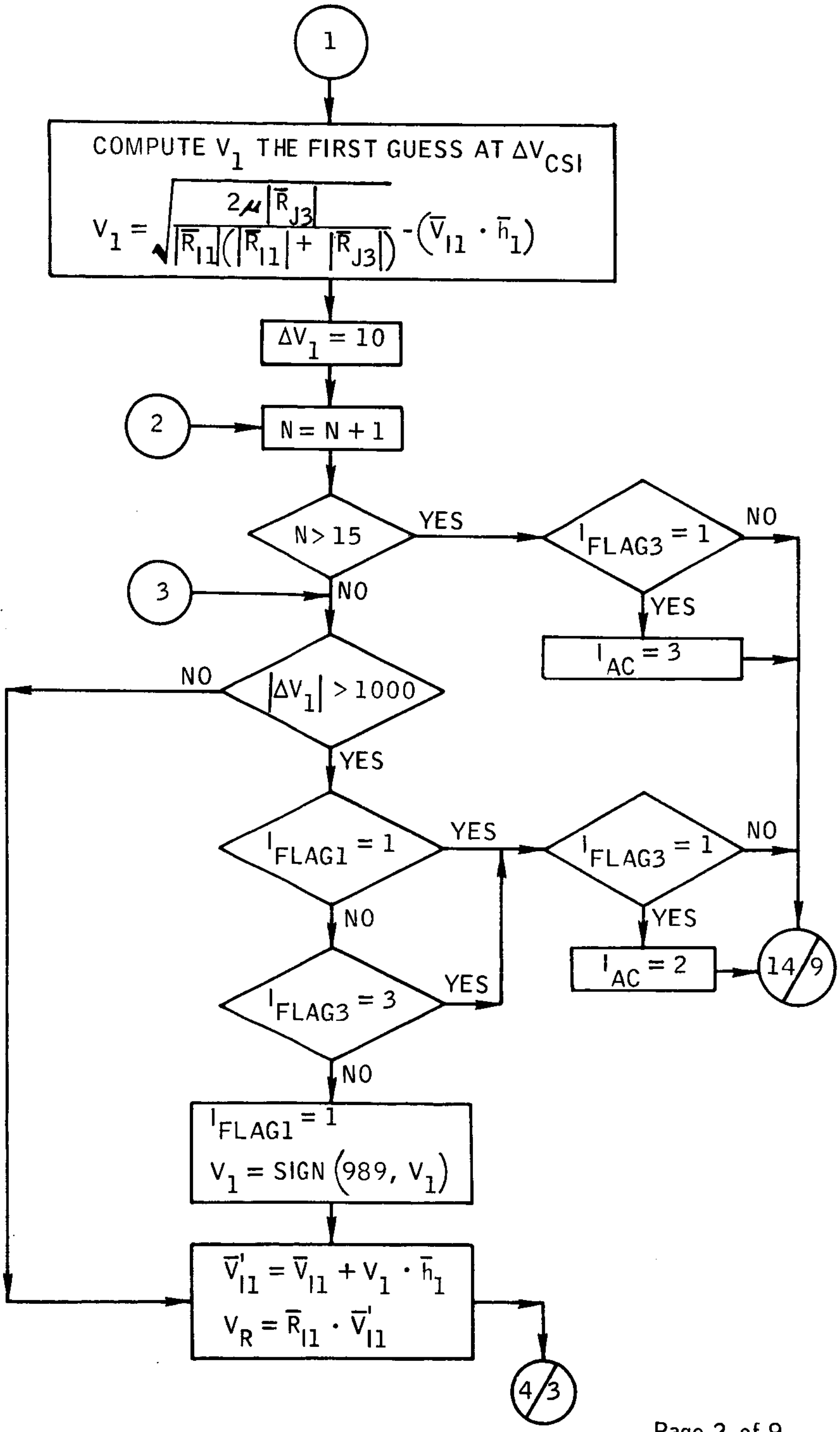
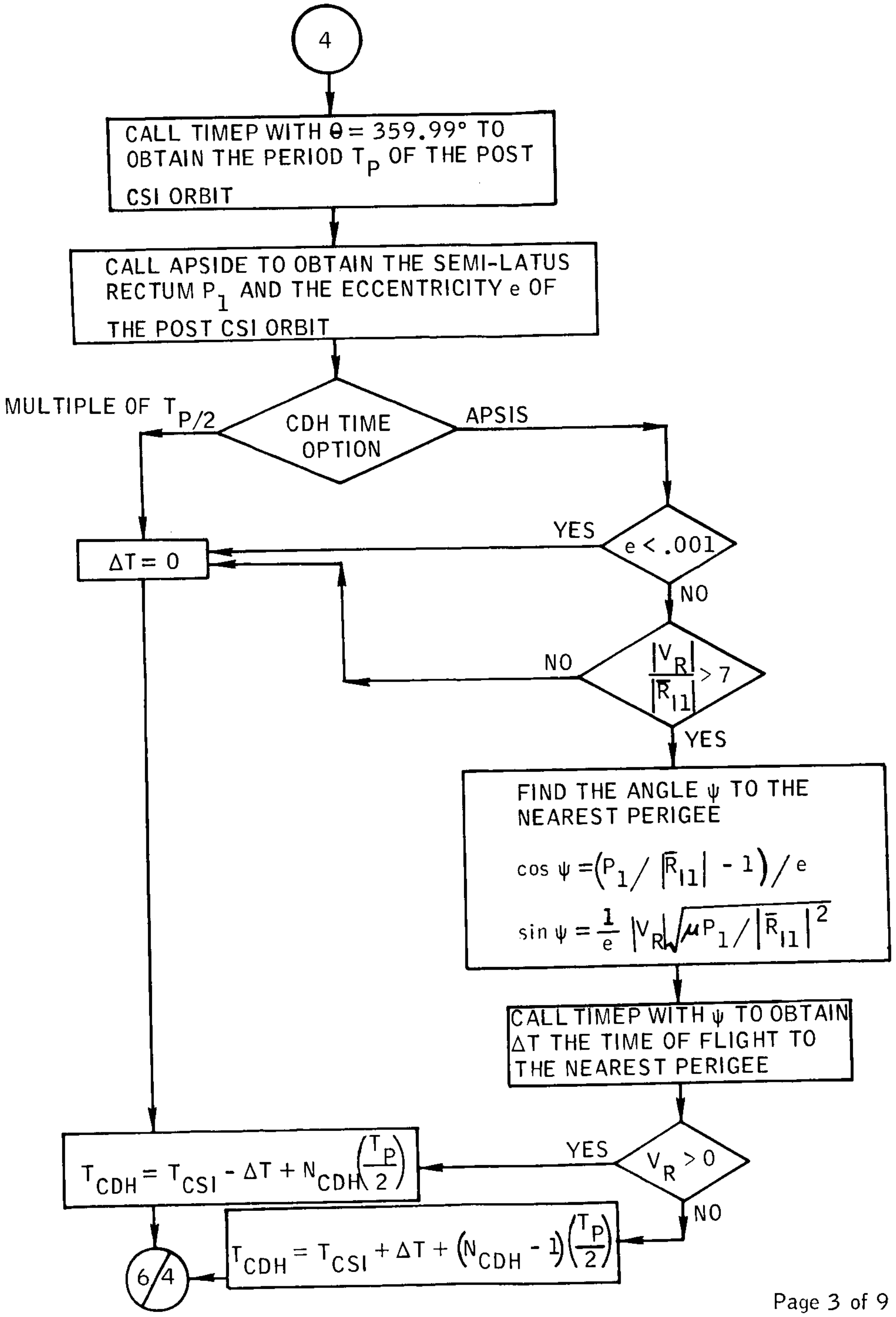
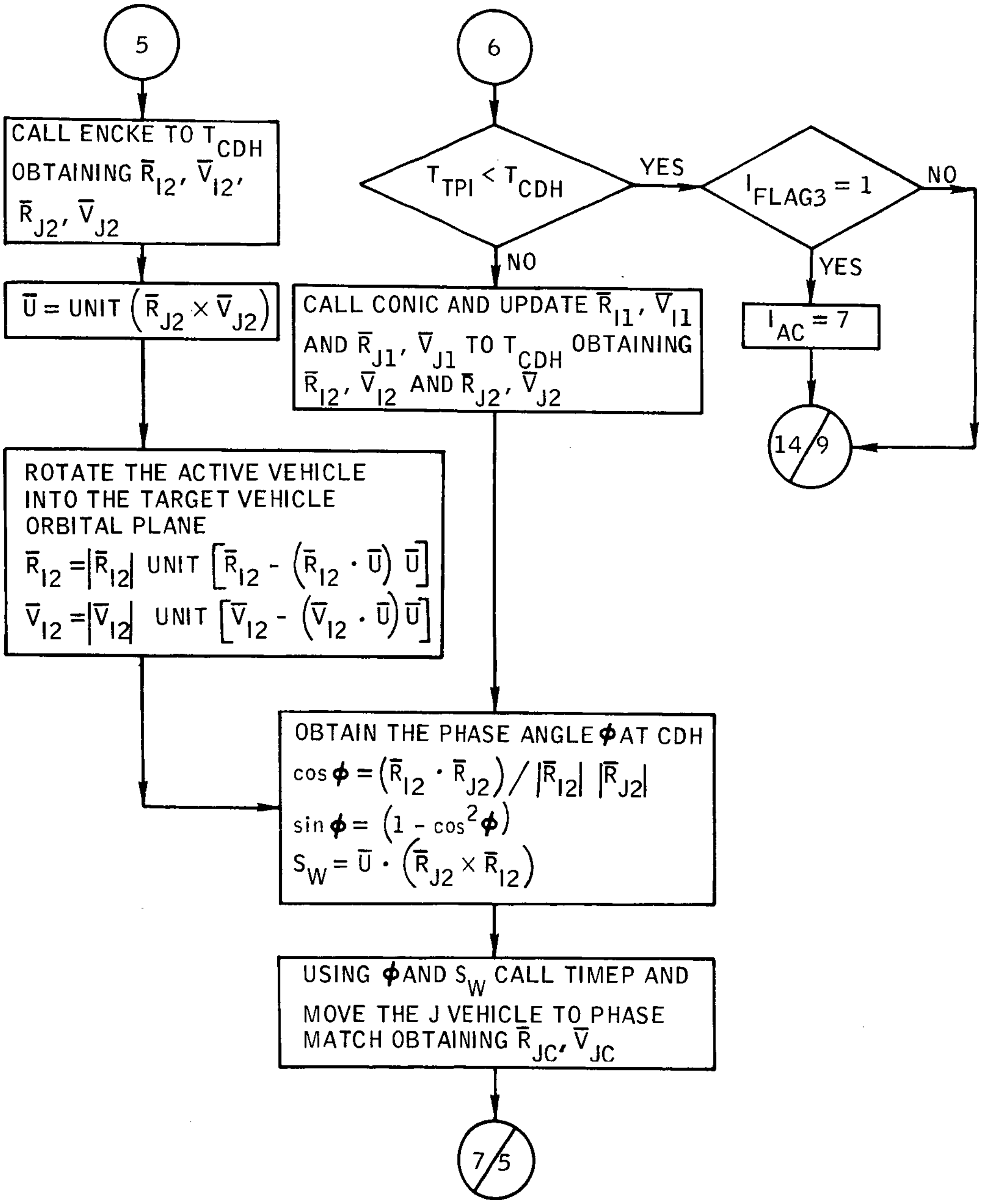

1
COMPUTE V_1 THE FIRST GUESS AT ΔV_{CSI}
$V_1 = \sqrt{\dfrac{2\mu|\bar{R}_{J3}|}{|\bar{R}_{I1}|\left(|\bar{R}_{I1}| + |\bar{R}_{J3}|\right)}} - \left(\bar{V}_{I1} \cdot \bar{h}_1\right)$
$\Delta V_1 = 10$
2
$N = N + 1$
$N > 15$
YES
$I_{FLAG3} = 1$
NO
YES
$I_{AC} = 3$
3
NO
$|\Delta V_1| > 1000$
NO
YES
$I_{FLAG1} = 1$
YES
$I_{FLAG3} = 1$
NO
NO
YES
$I_{AC} = 2$
14/9
$I_{FLAG3} = 3$
YES
NO
$I_{FLAG1} = 1$
$V_1 = \text{SIGN}(989, V_1)$
$\bar{V}'_{I1} = \bar{V}_{I1} + V_1 \cdot \bar{h}_1$
$V_R = \bar{R}_{I1} \cdot \bar{V}'_{I1}$
4/3

Figure B-18.- Continued.


4
CALL TIMEP WITH $\theta = 359.99°$ TO OBTAIN THE PERIOD T_P OF THE POST CSI ORBIT
CALL APSIDE TO OBTAIN THE SEMI-LATUS RECTUM P_1 AND THE ECCENTRICITY e OF THE POST CSI ORBIT
MULTIPLE OF $T_{P/2}$
CDH TIME OPTION
APSIS
YES
$e < .001$
$\Delta T = 0$
NO
NO
$\frac{|V_R|}{|\bar{R}_{I1}|} > 7$
YES
FIND THE ANGLE ψ TO THE NEAREST PERIGEE
$\cos\psi = \left(P_1 / |\bar{R}_{I1}| - 1\right) / e$
$\sin\psi = \frac{1}{e} |V_R| \sqrt{\mu P_1 / |\bar{R}_{I1}|^2}$
CALL TIMEP WITH ψ TO OBTAIN ΔT THE TIME OF FLIGHT TO THE NEAREST PERIGEE
YES
$V_R > 0$
NO
$T_{CDH} = T_{CSI} - \Delta T + N_{CDH}\left(\frac{T_P}{2}\right)$
$T_{CDH} = T_{CSI} + \Delta T + \left(N_{CDH} - 1\right)\left(\frac{T_P}{2}\right)$
6/4

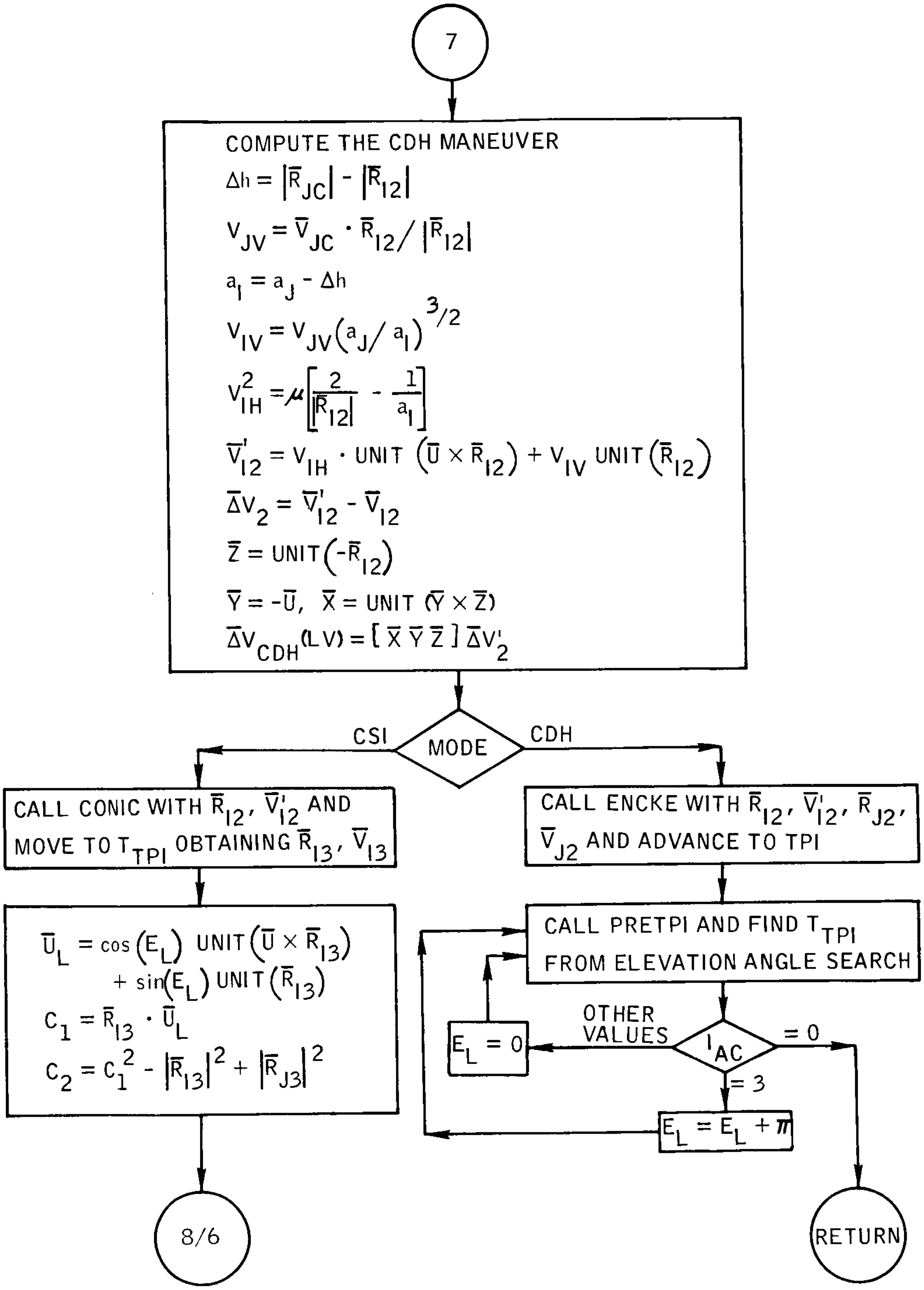
Page 3 of 9

Figure B-18.- Continued.


5
CALL ENCKE TO T_{CDH} OBTAINING $\bar{R}_{I2}$, $\bar{V}_{I2}$, $\bar{R}_{J2}$, $\bar{V}_{J2}$
$\bar{U} = \text{UNIT}\left(\bar{R}_{J2} \times \bar{V}_{J2}\right)$
ROTATE THE ACTIVE VEHICLE INTO THE TARGET VEHICLE ORBITAL PLANE
$\bar{R}_{I2} = \left|\bar{R}_{I2}\right| \text{ UNIT } \left[\bar{R}_{I2} - \left(\bar{R}_{I2} \cdot \bar{U}\right) \bar{U}\right]$
$\bar{V}_{I2} = \left|\bar{V}_{I2}\right| \text{ UNIT } \left[\bar{V}_{I2} - \left(\bar{V}_{I2} \cdot \bar{U}\right) \bar{U}\right]$
6
$T_{TPI} < T_{CDH}$
YES
NO
$I_{FLAG3} = 1$
NO
YES
$I_{AC} = 7$
14/9
CALL CONIC AND UPDATE $\bar{R}_{I1}$, $\bar{V}_{I1}$ AND $\bar{R}_{J1}$, $\bar{V}_{J1}$ TO T_{CDH} OBTAINING $\bar{R}_{I2}$, $\bar{V}_{I2}$ AND $\bar{R}_{J2}$, $\bar{V}_{J2}$
OBTAIN THE PHASE ANGLE ϕ AT CDH
$\cos\phi = \left(\bar{R}_{I2} \cdot \bar{R}_{J2}\right) / \left|\bar{R}_{I2}\right| \left|\bar{R}_{J2}\right|$
$\sin\phi = \left(1 - \cos^2\phi\right)$
$S_W = \bar{U} \cdot \left(\bar{R}_{J2} \times \bar{R}_{I2}\right)$
USING ϕ AND S_W CALL TIMEP AND MOVE THE J VEHICLE TO PHASE MATCH OBTAINING $\bar{R}_{JC}$, $\bar{V}_{JC}$
7/5

Page 4 of 9

Figure B-18.- Continued.


7
COMPUTE THE CDH MANEUVER
$\Delta h = |\bar{R}_{JC}| - |\bar{R}_{12}|$
$V_{JV} = \bar{V}_{JC} \cdot \bar{R}_{12} / |\bar{R}_{12}|$
$a_I = a_J - \Delta h$
$V_{IV} = V_{JV}(a_J / a_I)^{3/2}$
$V_{IH}^2 = \mu \left[\frac{2}{|\bar{R}_{12}|} - \frac{1}{a_I} \right]$
$\bar{V}'_{12} = V_{IH} \cdot \text{UNIT}\left(\bar{U} \times \bar{R}_{12}\right) + V_{IV}\ \text{UNIT}\left(\bar{R}_{12}\right)$
$\bar{\Delta V}_2 = \bar{V}'_{12} - \bar{V}_{12}$
$\bar{Z} = \text{UNIT}\left(-\bar{R}_{12}\right)$
$\bar{Y} = -\bar{U}, \quad \bar{X} = \text{UNIT}\ (\bar{Y} \times \bar{Z})$
$\bar{\Delta V}_{CDH}(LV) = [\ \bar{X}\ \bar{Y}\ \bar{Z}\ ]\ \bar{\Delta V}'_2$
CSI
MODE
CDH
CALL CONIC WITH $\bar{R}_{12}$, $\bar{V}'_{12}$ AND MOVE TO T_{TPI} OBTAINING $\bar{R}_{13}$, $\bar{V}_{13}$
CALL ENCKE WITH $\bar{R}_{12}$, $\bar{V}'_{12}$, $\bar{R}_{J2}$, $\bar{V}_{J2}$ AND ADVANCE TO TPI
$\bar{U}_L = \cos\left(E_L\right)\ \text{UNIT}\left(\bar{U} \times \bar{R}_{13}\right) + \sin\left(E_L\right)\ \text{UNIT}\left(\bar{R}_{13}\right)$
$C_1 = \bar{R}_{13} \cdot \bar{U}_L$
$C_2 = C_1^2 - |\bar{R}_{13}|^2 + |\bar{R}_{J3}|^2$
CALL PRETPI AND FIND T_{TPI} FROM ELEVATION ANGLE SEARCH
$E_L = 0$
OTHER VALUES
I_{AC}
= 0
= 3
$E_L = E_L + \pi$
8/6
RETURN

Page 5 of 9

Figure B-18.- Continued.

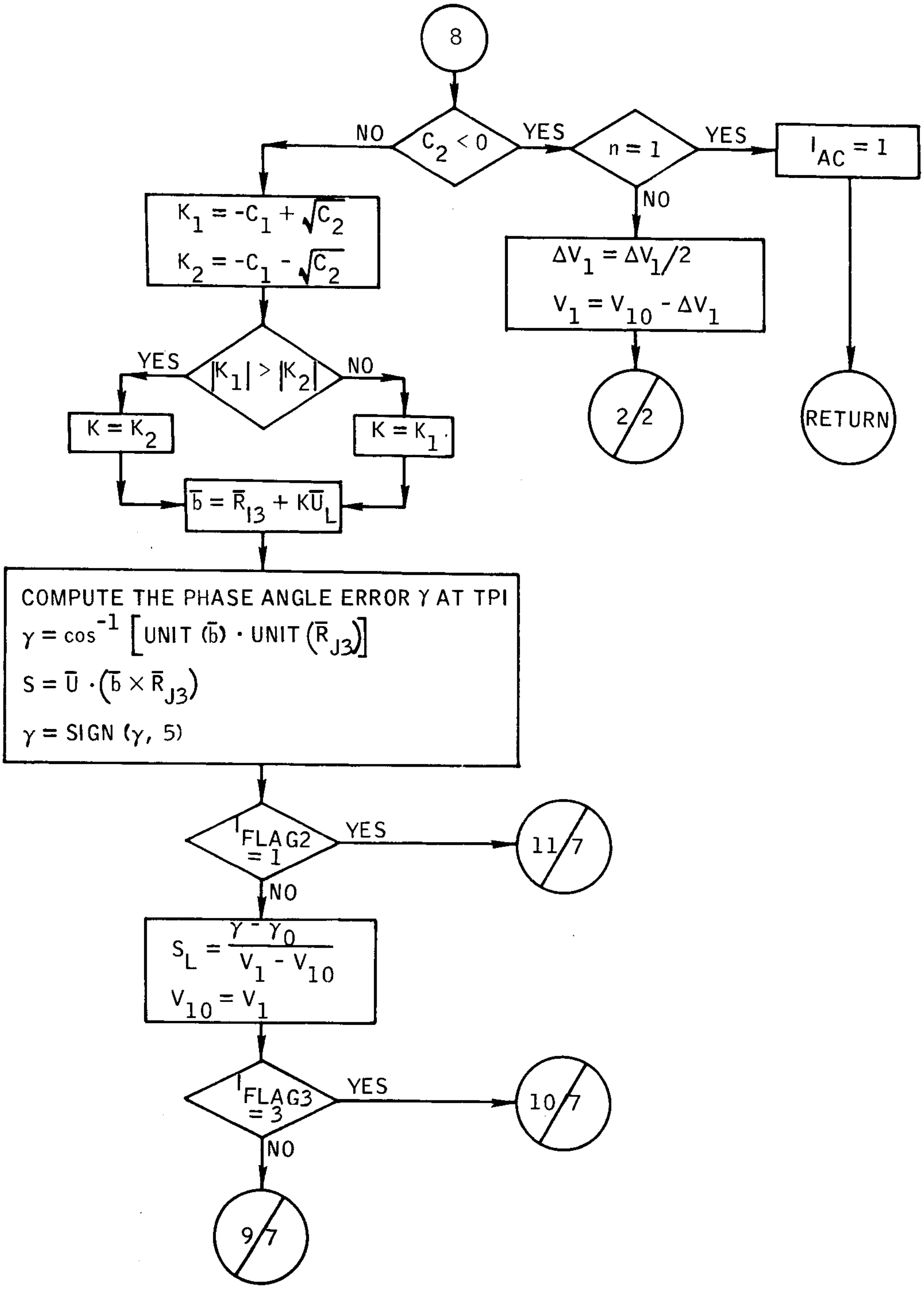
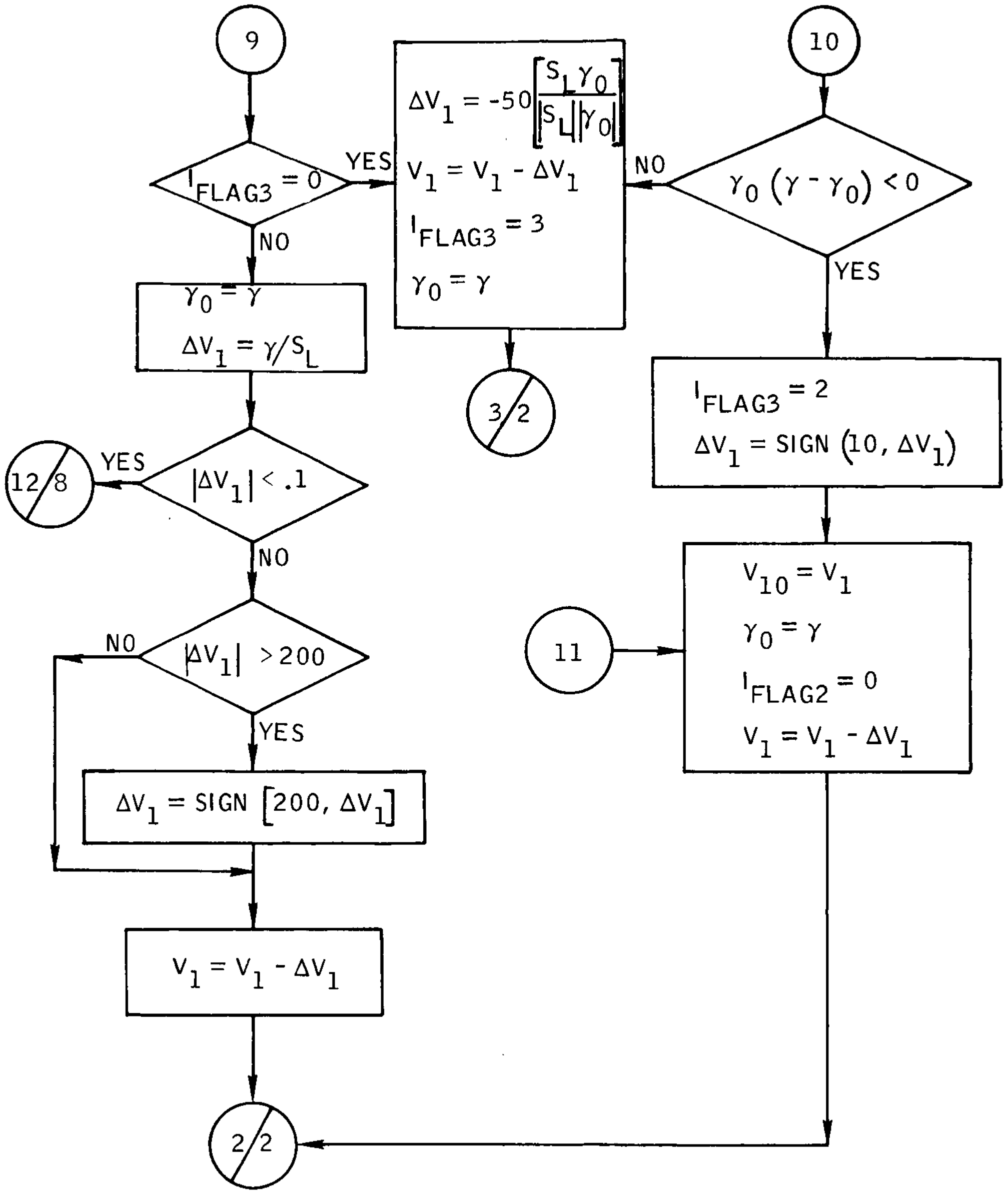

8
$C_2 < 0$
NO
YES
$n = 1$
YES
$I_{AC} = 1$
NO
$K_1 = -C_1 + \sqrt{C_2}$
$K_2 = -C_1 - \sqrt{C_2}$
$\Delta V_1 = \Delta V_1/2$
$V_1 = V_{10} - \Delta V_1$
$|K_1| > |K_2|$
YES
NO
$K = K_2$
$K = K_1$
2/2
RETURN
$\bar{b} = \bar{R}_{I3} + K\bar{U}_L$
COMPUTE THE PHASE ANGLE ERROR γ AT TPI
$\gamma = \cos^{-1}\left[\text{UNIT}(\bar{b}) \cdot \text{UNIT}(\bar{R}_{J3})\right]$
$S = \bar{U} \cdot (\bar{b} \times \bar{R}_{J3})$
$\gamma = \text{SIGN}(\gamma, S)$
$I_{FLAG2} = 1$
YES
11/7
NO
$S_L = \dfrac{\gamma - \gamma_0}{V_1 - V_{10}}$
$V_{10} = V_1$
$I_{FLAG3} = 3$
YES
10/7
NO
9/7

Figure B-18.- Continued.



Page 7 of 9

Figure B-18.- Continued.

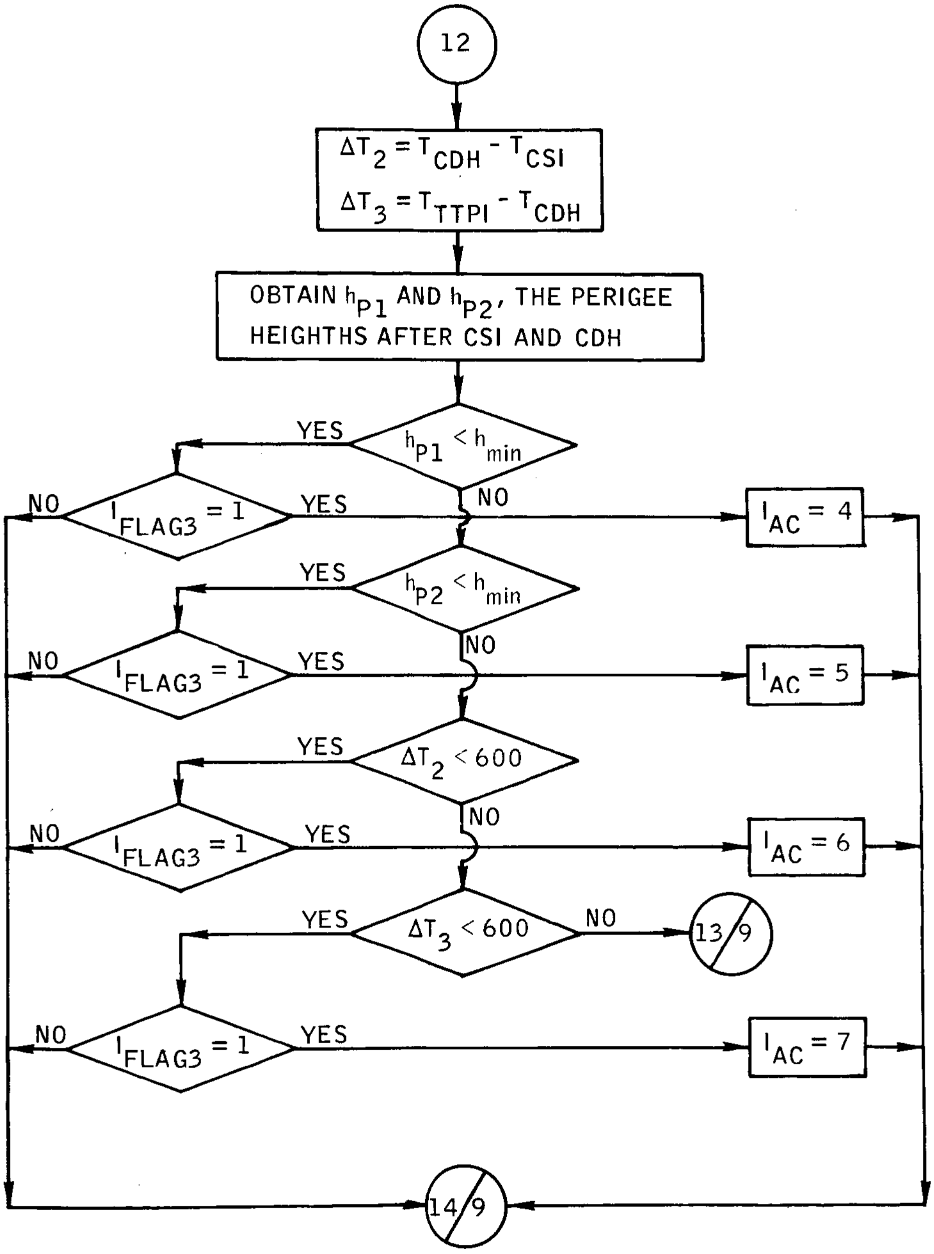
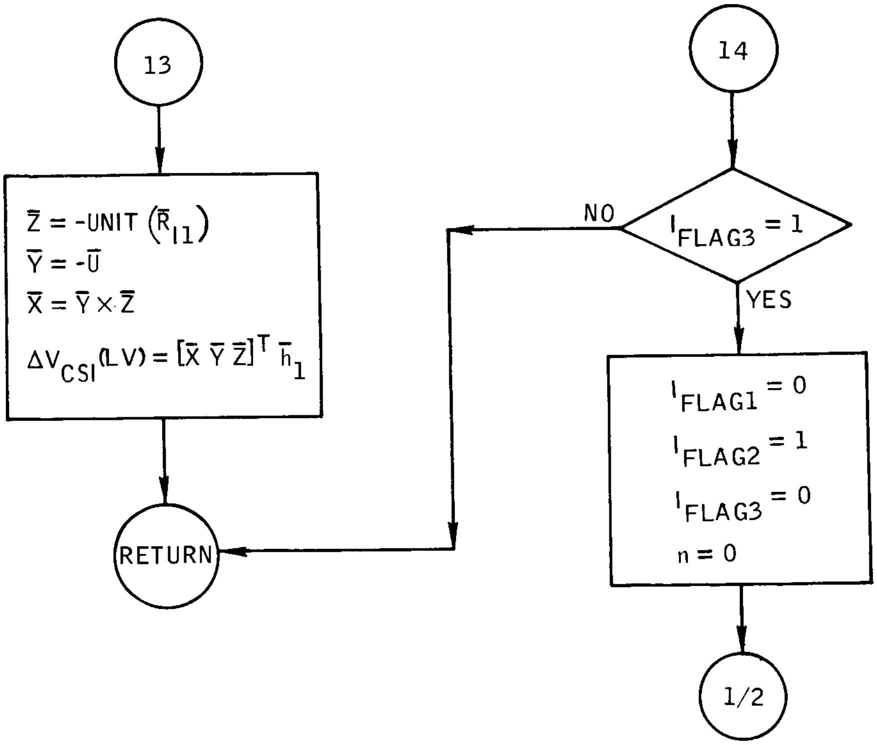


Figure B-18.- Continued.


13
$\bar{Z} = -\text{UNIT}\left(\bar{R}_{I1}\right)$
$\bar{Y} = -\bar{U}$
$\bar{X} = \bar{Y} \times \bar{Z}$
$\Delta V_{CSI}(LV) = [\bar{X}\ \bar{Y}\ \bar{Z}]^T\ \bar{h}_1$
RETURN
14
NO
$I_{FLAG3} = 1$
YES
$I_{FLAG1} = 0$
$I_{FLAG2} = 1$
$I_{FLAG3} = 0$
$n = 0$
1/2

Page 9 of 9

Figure B-18.- Concluded.

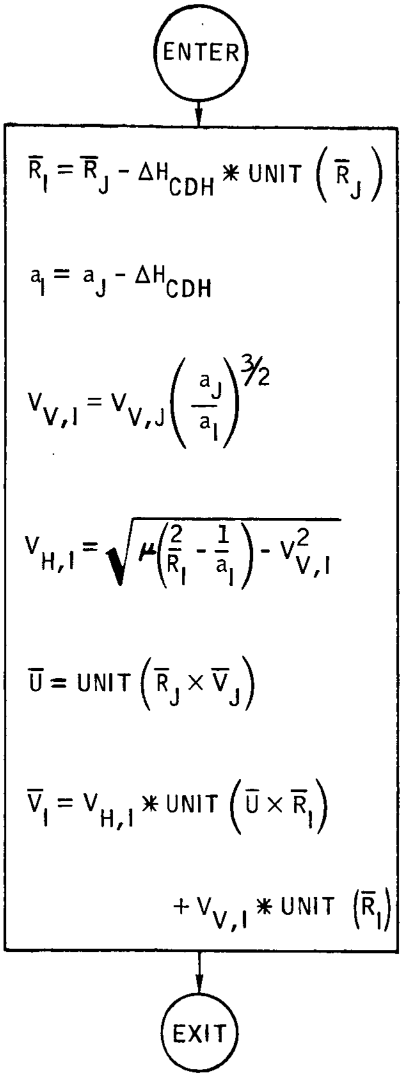
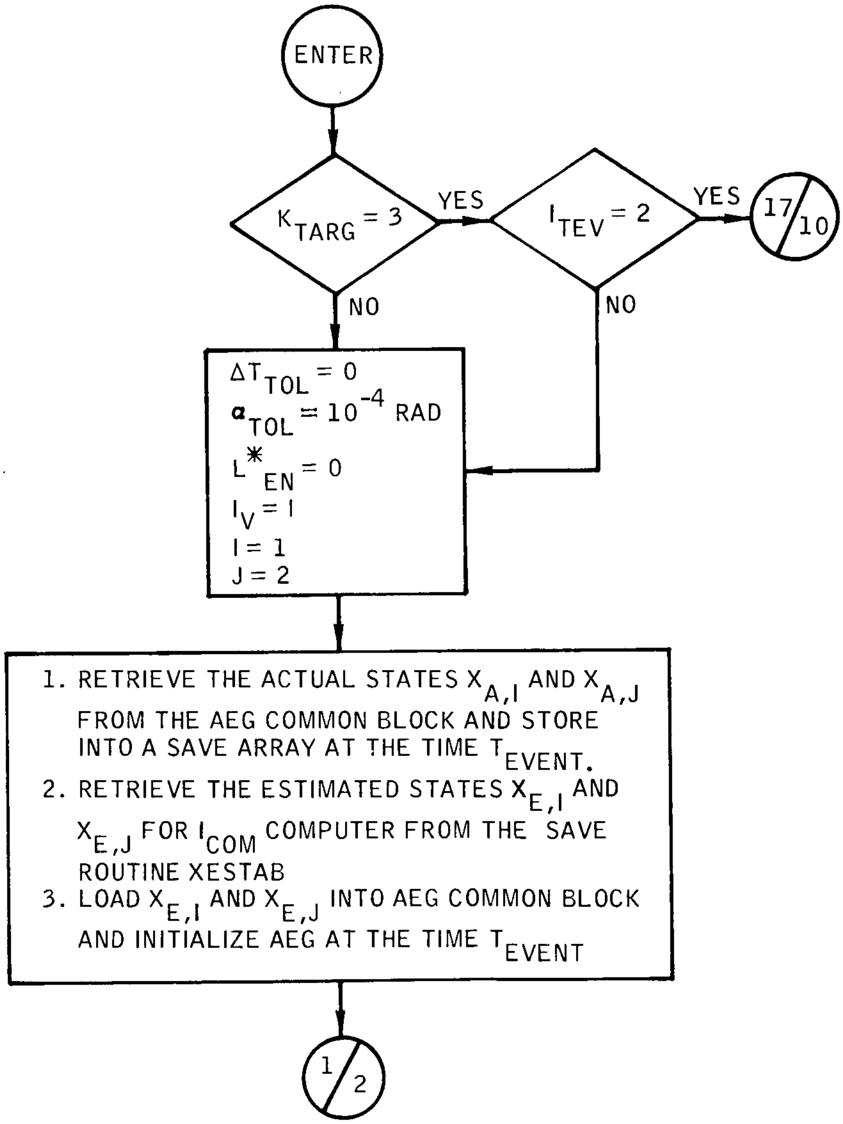

ENTER
$\bar{R}_I = \bar{R}_J - \Delta H_{CDH} * \text{UNIT}\left(\bar{R}_J\right)$
$a_I = a_J - \Delta H_{CDH}$
$V_{V,I} = V_{V,J}\left(\frac{a_J}{a_I}\right)^{3/2}$
$V_{H,I} = \sqrt{\mu\left(\frac{2}{R_I} - \frac{1}{a_I}\right) - V_{V,I}^2}$
$\bar{U} = \text{UNIT}\left(\bar{R}_J \times \bar{V}_J\right)$
$\bar{V}_I = V_{H,I} * \text{UNIT}\left(\bar{U} \times \bar{R}_I\right)$
$+ V_{V,I} * \text{UNIT}\left(\bar{R}_I\right)$
EXIT

Figure B-19.- Flow chart of subroutine COE.

Page 1 of 1


ENTER
$K_{TARG} = 3$
YES
$I_{TEV} = 2$
YES
17/10
NO
NO
$\Delta T_{TOL} = 0$
$\alpha_{TOL} = 10^{-4}$ RAD
$L^*_{EN} = 0$
$I_V = 1$
$I = 1$
$J = 2$
1. RETRIEVE THE ACTUAL STATES $X_{A,I}$ AND $X_{A,J}$ FROM THE AEG COMMON BLOCK AND STORE INTO A SAVE ARRAY AT THE TIME T_{EVENT}.
2. RETRIEVE THE ESTIMATED STATES $X_{E,I}$ AND $X_{E,J}$ FOR I_{COM} COMPUTER FROM THE SAVE ROUTINE XESTAB
3. LOAD $X_{E,I}$ AND $X_{E,J}$ INTO AEG COMMON BLOCK AND INITIALIZE AEG AT THE TIME T_{EVENT}
1/2

Page 1 of 10

Figure B-20.- Flow chart of subroutine COMDKI.

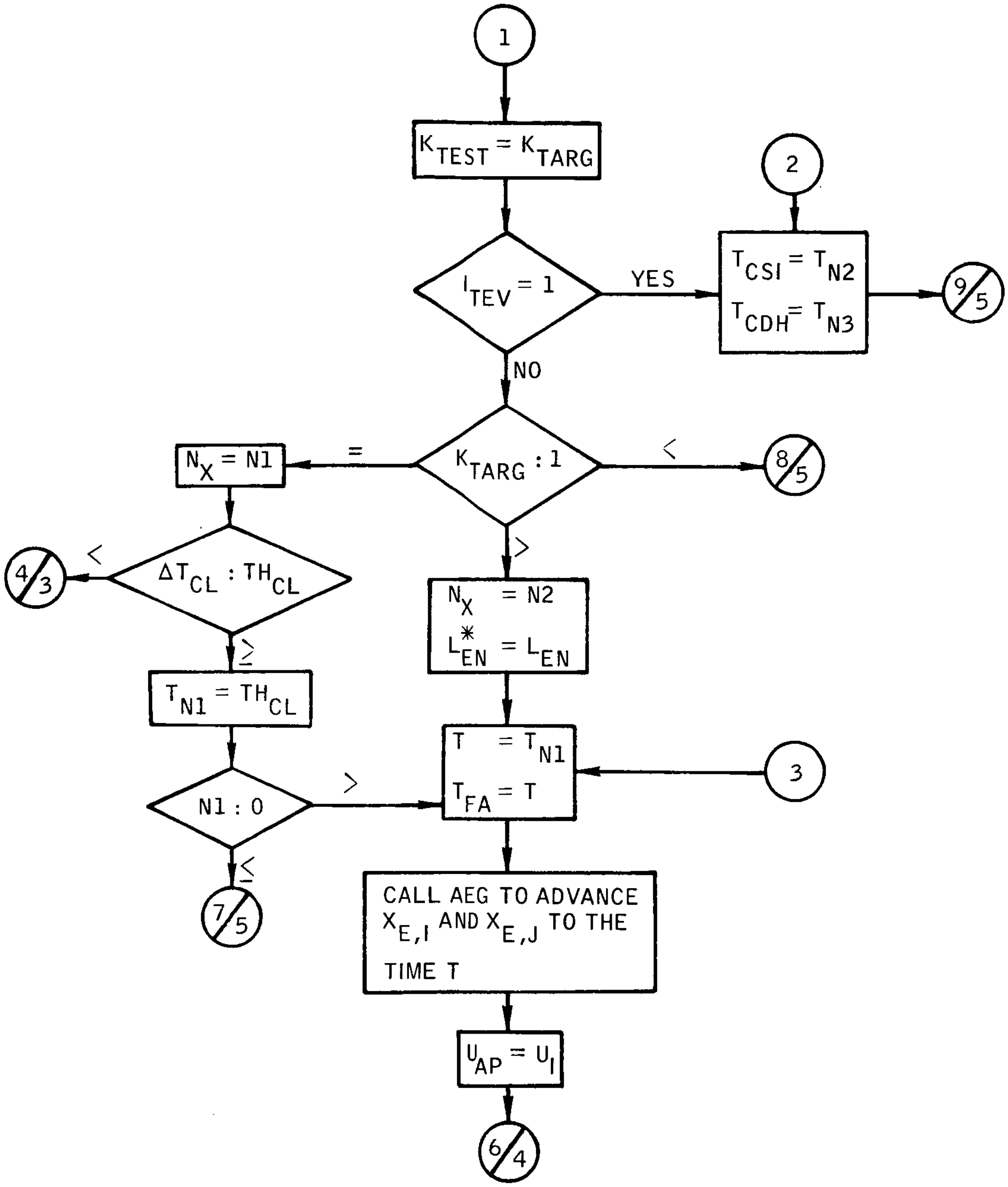
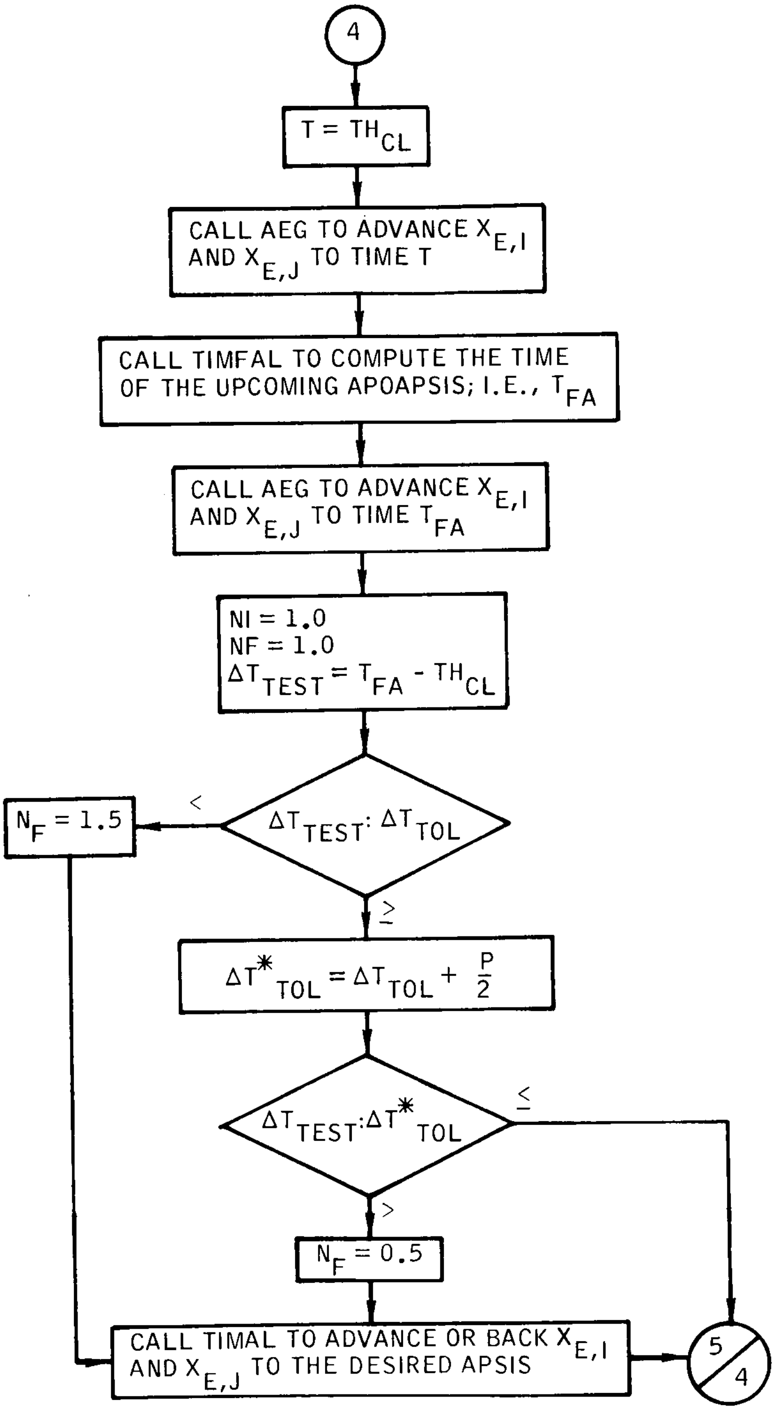


Figure B-20.- Continued.

Page 2 of 10


4
$T = TH_{CL}$
CALL AEG TO ADVANCE $X_{E,I}$ AND $X_{E,J}$ TO TIME T
CALL TIMFAL TO COMPUTE THE TIME OF THE UPCOMING APOAPSIS; I.E., T_{FA}
CALL AEG TO ADVANCE $X_{E,I}$ AND $X_{E,J}$ TO TIME T_{FA}
NI = 1.0
NF = 1.0
$\Delta T_{TEST} = T_{FA} - TH_{CL}$
$\Delta T_{TEST} : \Delta T_{TOL}$
<
$N_F = 1.5$
≥
$\Delta T^{*}_{TOL} = \Delta T_{TOL} + \frac{P}{2}$
$\Delta T_{TEST} : \Delta T^{*}_{TOL}$
≤
>
$N_F = 0.5$
CALL TIMAL TO ADVANCE OR BACK $X_{E,I}$ AND $X_{E,J}$ TO THE DESIRED APSIS
5
4

Page 3 of 10

Figure B-20.- Continued.

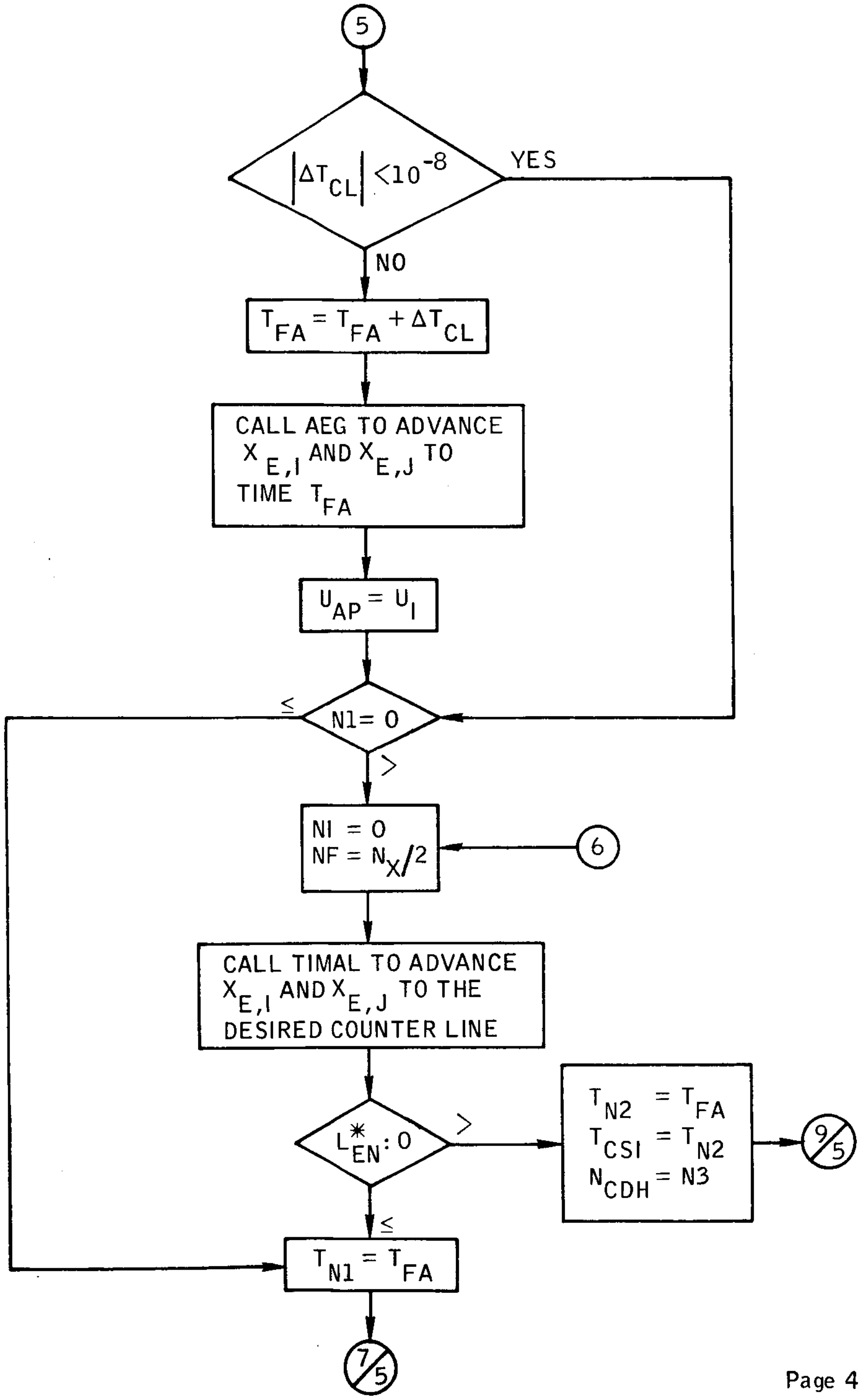
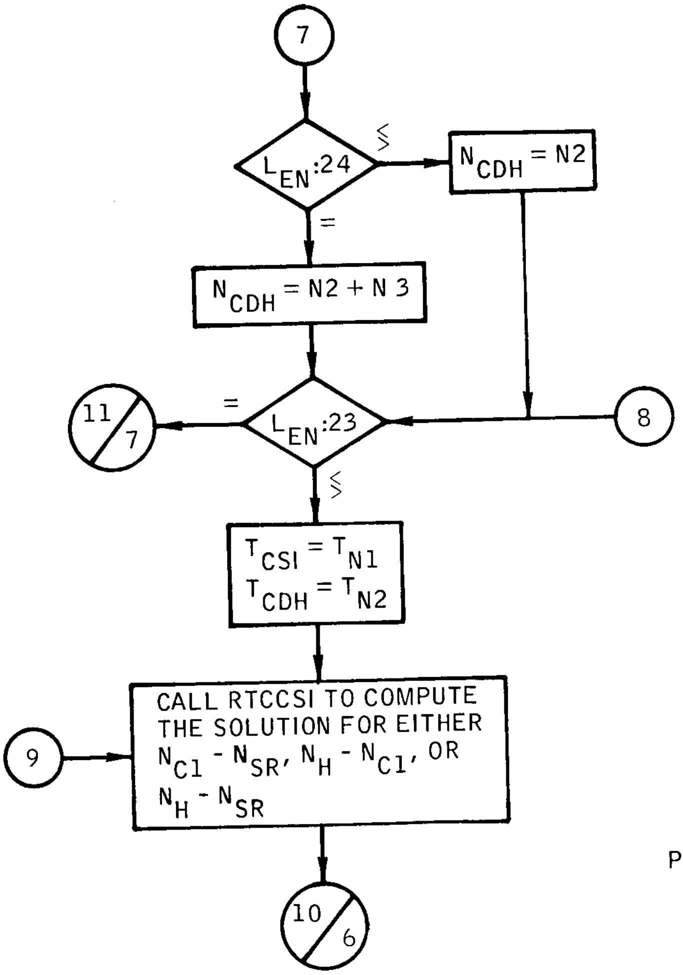

5
$|\Delta T_{CL}| < 10^{-8}$
YES
NO
$T_{FA} = T_{FA} + \Delta T_{CL}$
CALL AEG TO ADVANCE $X_{E,I}$ AND $X_{E,J}$ TO TIME T_{FA}
$U_{AP} = U_I$
$N1 = 0$
≤
>
$NI = 0$
$NF = N_X/2$
6
CALL TIMAL TO ADVANCE $X_{E,I}$ AND $X_{E,J}$ TO THE DESIRED COUNTER LINE
$L^*_{EN} : 0$
>
≤
$T_{N2} = T_{FA}$
$T_{CSI} = T_{N2}$
$N_{CDH} = N3$
9/5
$T_{N1} = T_{FA}$
7/5

Figure B-20.- Continued.



Page 5 of 10

Figure B-20.- Continued.

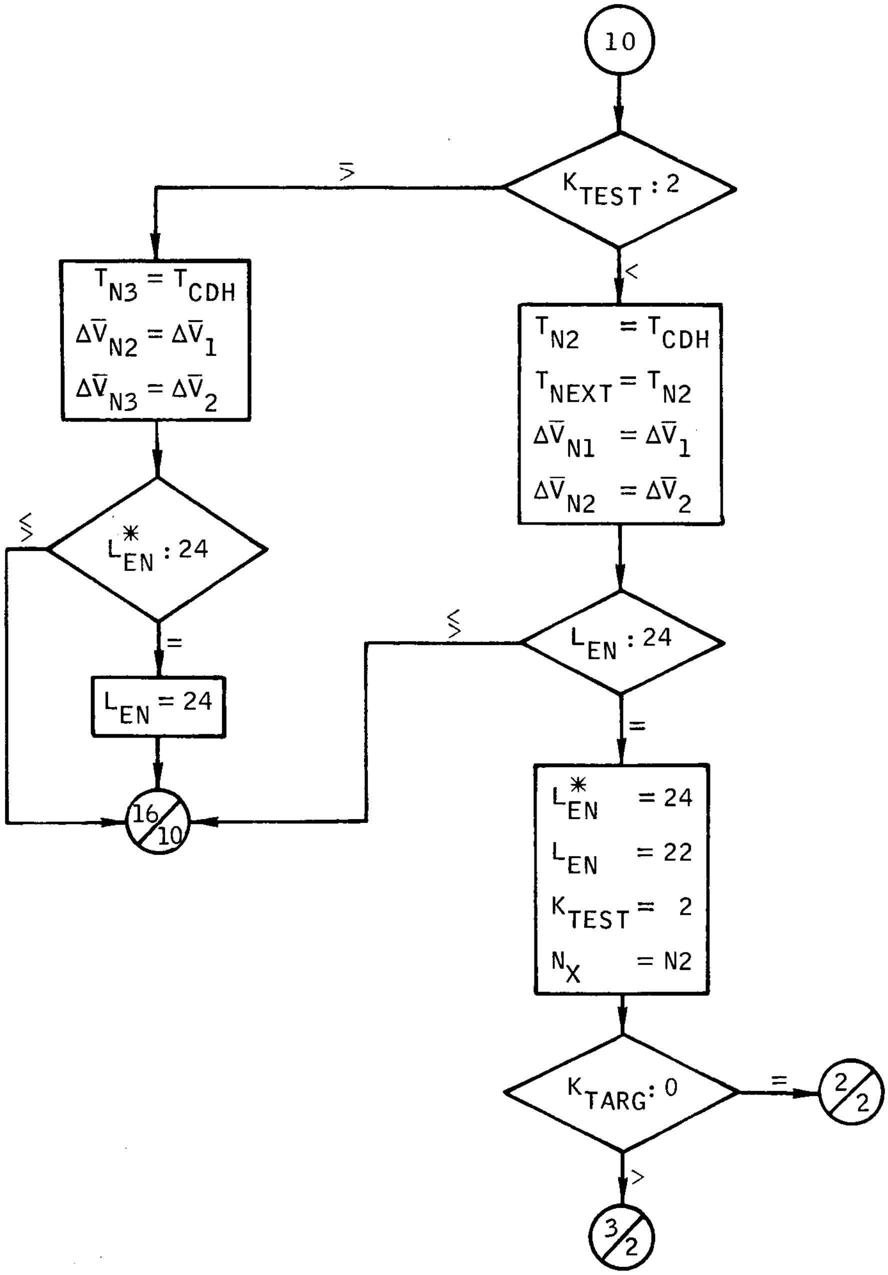
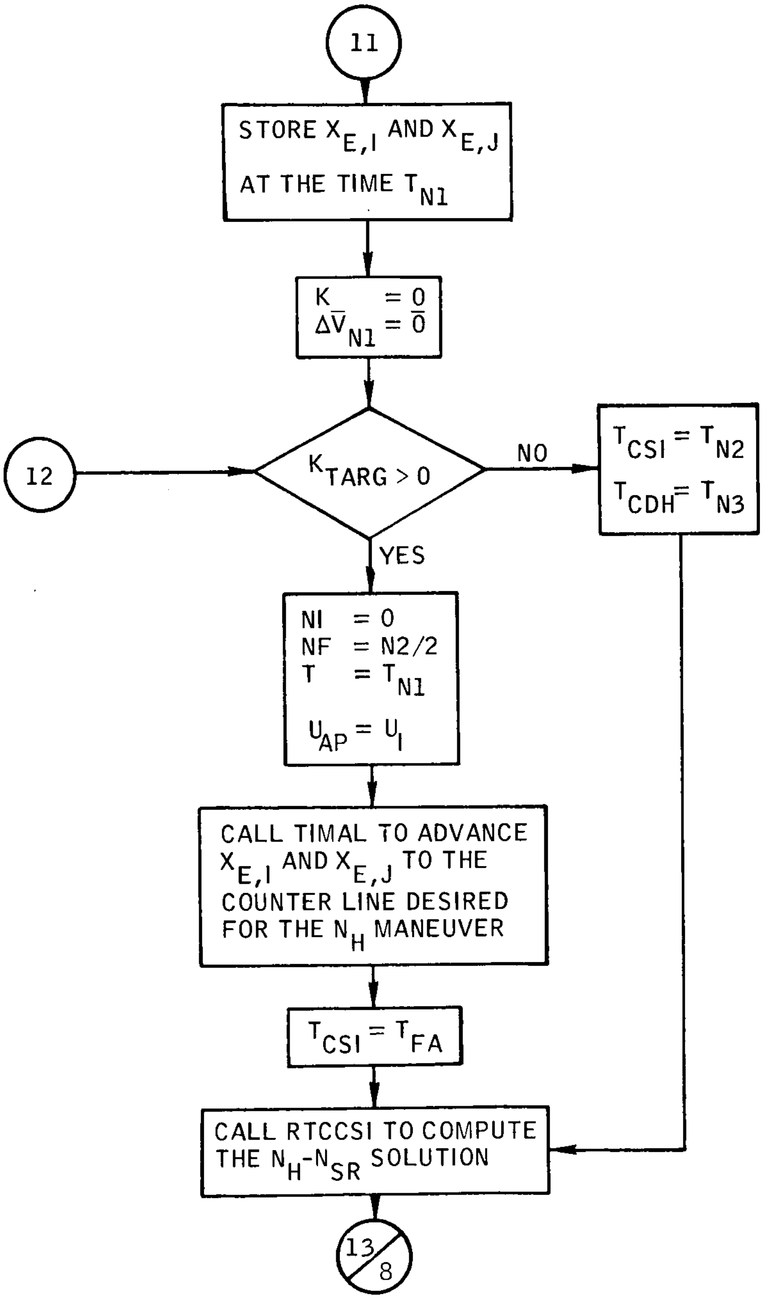


Figure B-20.- Continued.


11
STORE $X_{E,I}$ AND $X_{E,J}$ AT THE TIME T_{N1}
$K = 0$
$\Delta \bar{V}_{N1} = \bar{0}$
12
$K_{TARG} > 0$
NO
$T_{CSI} = T_{N2}$
$T_{CDH} = T_{N3}$
YES
NI = 0
NF = N2/2
$T = T_{N1}$
$U_{AP} = U_I$
CALL TIMAL TO ADVANCE $X_{E,I}$ AND $X_{E,J}$ TO THE COUNTER LINE DESIRED FOR THE N_H MANEUVER
$T_{CSI} = T_{FA}$
CALL RTCCSI TO COMPUTE THE N_H-N_{SR} SOLUTION
13/8

Page 7 of 10

Figure B-20.- Continued.

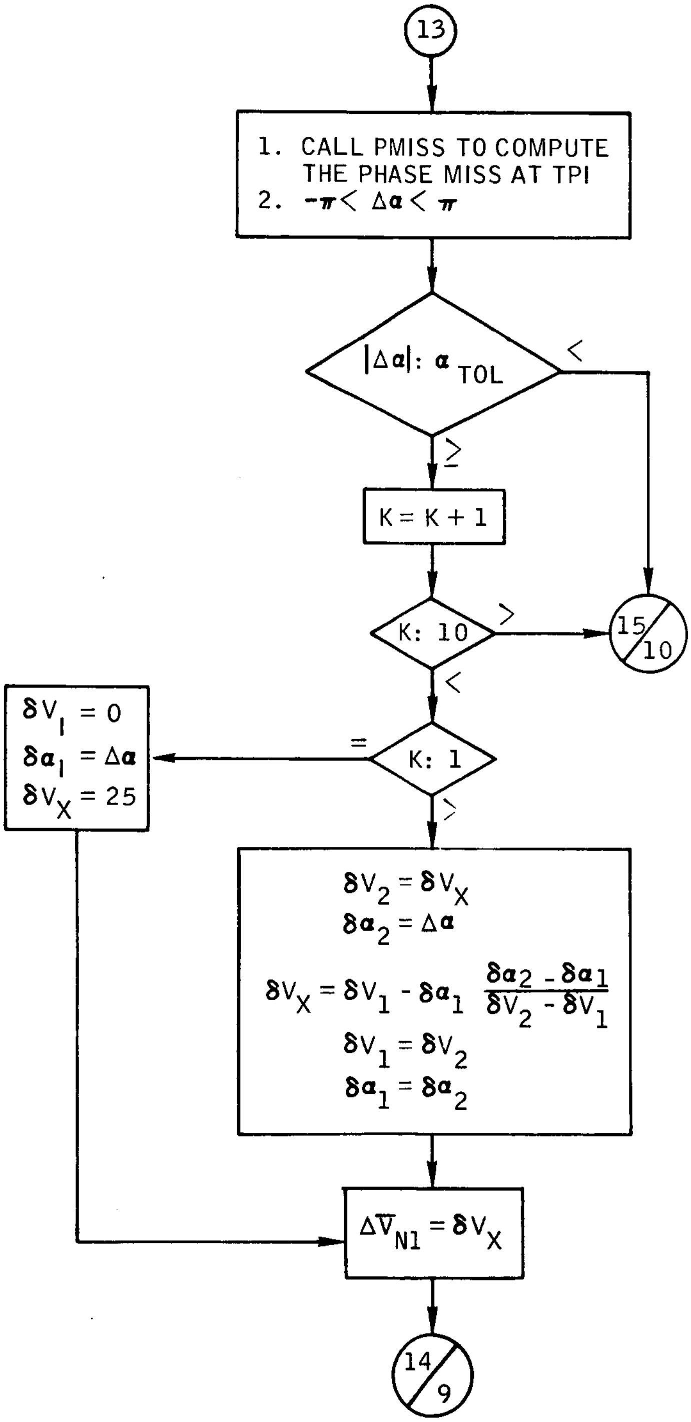
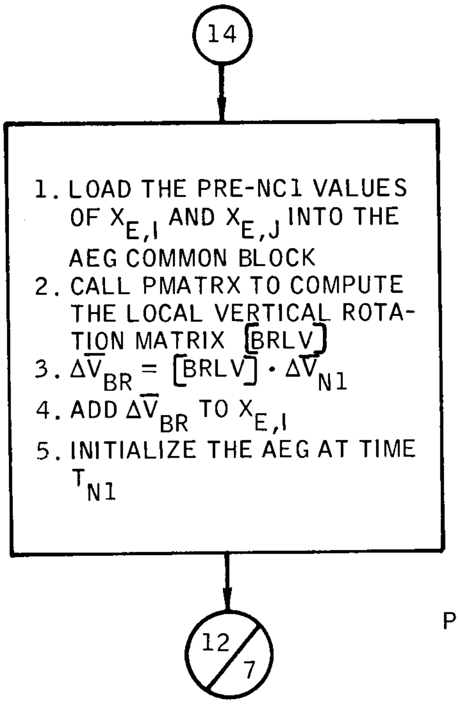

13
1. CALL PMISS TO COMPUTE THE PHASE MISS AT TPI
2. $-\pi < \Delta\alpha < \pi$
$|\Delta\alpha| : \alpha_{TOL}$
<
$\geq$
K = K + 1
K: 10
>
15/10
<
K: 1
=
$\delta V_1 = 0$
$\delta\alpha_1 = \Delta\alpha$
$\delta V_X = 25$
>
$\delta V_2 = \delta V_X$
$\delta\alpha_2 = \Delta\alpha$
$\delta V_X = \delta V_1 - \delta\alpha_1 \frac{\delta\alpha_2 - \delta\alpha_1}{\delta V_2 - \delta V_1}$
$\delta V_1 = \delta V_2$
$\delta\alpha_1 = \delta\alpha_2$
$\Delta\overline{V}_{N1} = \delta V_X$
14/9

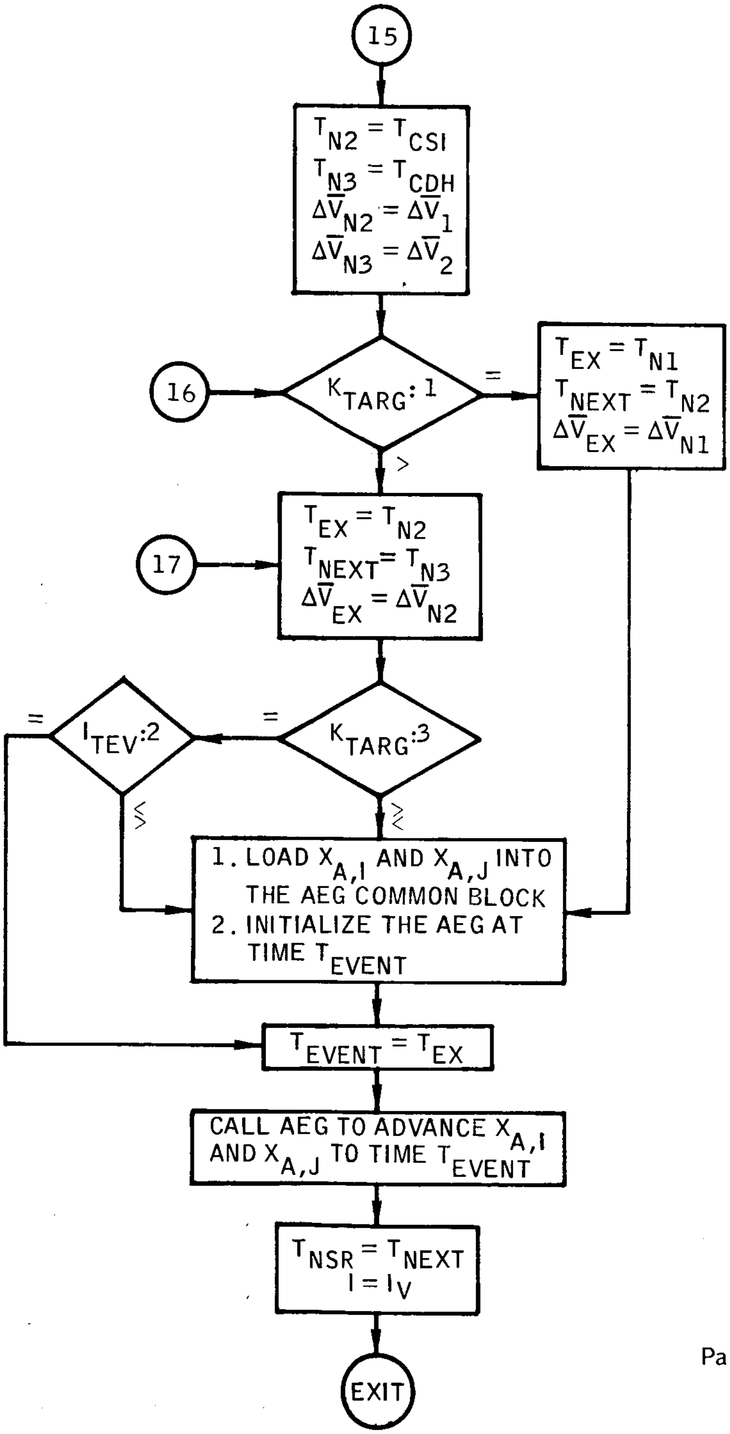
Figure B-20.- Continued.



1. LOAD THE PRE-NC1 VALUES OF $X_{E,I}$ AND $X_{E,J}$ INTO THE AEG COMMON BLOCK
2. CALL PMATRX TO COMPUTE THE LOCAL VERTICAL ROTATION MATRIX $\left[\underline{BRLV}\right]$
3. $\Delta\overline{V}_{BR} = \left[\overline{BRLV}\right] \cdot \Delta\overline{V}_{N1}$
4. ADD $\Delta\overline{V}_{BR}$ TO $X_{E,I}$
5. INITIALIZE THE AEG AT TIME T_{N1}

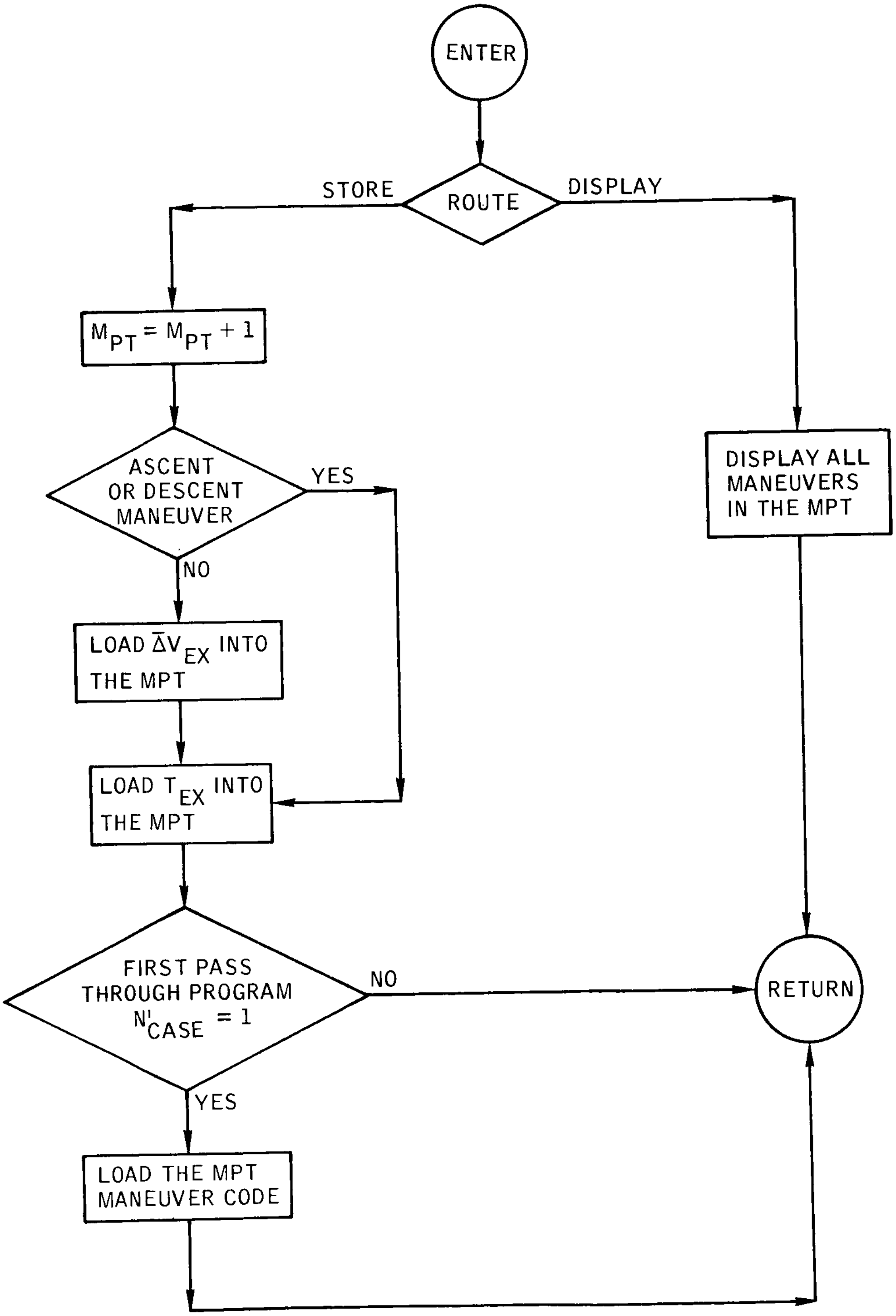
Page 9 of 10

Figure B-20.- Continued.


15
$T_{N2} = T_{CSI}$
$T_{N3} = T_{CDH}$
$\Delta\overline{V}_{N2} = \Delta\overline{V}_1$
$\Delta\overline{V}_{N3} = \Delta\overline{V}_2$
16
$K_{TARG}:1$
=
$T_{EX} = T_{N1}$
$T_{NEXT} = T_{N2}$
$\Delta\overline{V}_{EX} = \Delta\overline{V}_{N1}$
>
17
$T_{EX} = T_{N2}$
$T_{NEXT} = T_{N3}$
$\Delta\overline{V}_{EX} = \Delta\overline{V}_{N2}$
=
$I_{TEV}:2$
=
$K_{TARG}:3$
$\lessgtr$
$\gtrless$
1. LOAD $X_{A,I}$ AND $X_{A,J}$ INTO THE AEG COMMON BLOCK
2. INITIALIZE THE AEG AT TIME T_{EVENT}
$T_{EVENT} = T_{EX}$
CALL AEG TO ADVANCE $X_{A,I}$ AND $X_{A,J}$ TO TIME T_{EVENT}
$T_{NSR} = T_{NEXT}$
$I = I_V$
EXIT

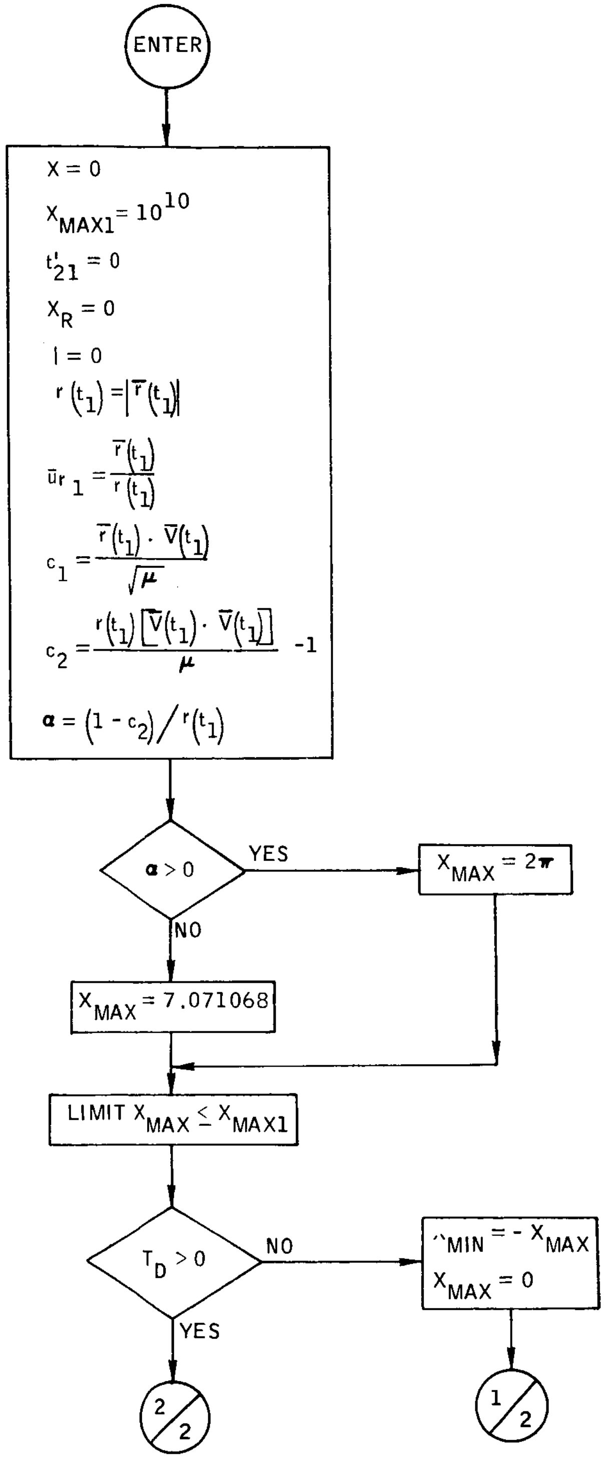
Page 10 of 10

Figure B-20.- Concluded.


ENTER
ROUTE
STORE
DISPLAY
$M_{PT} = M_{PT} + 1$
ASCENT OR DESCENT MANEUVER
YES
NO
LOAD $\bar{\Delta}V_{EX}$ INTO THE MPT
LOAD T_{EX} INTO THE MPT
FIRST PASS THROUGH PROGRAM $N'_{CASE} = 1$
NO
YES
LOAD THE MPT MANEUVER CODE
DISPLAY ALL MANEUVERS IN THE MPT
RETURN

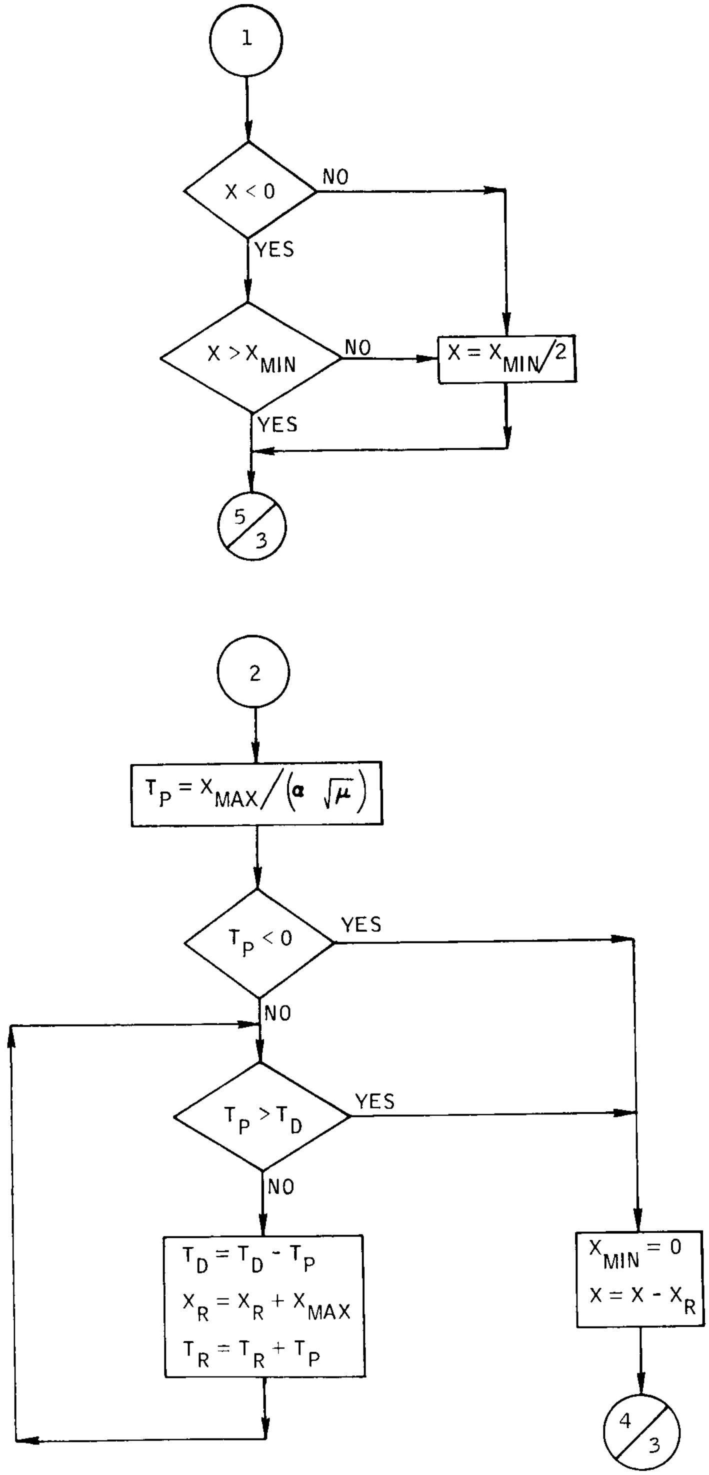
Page 1 of 1

Figure B-21.- Flow chart of subroutine COMMPT.


ENTER
$X = 0$
$X_{MAX1} = 10^{10}$
$t'_{21} = 0$
$X_R = 0$
$I = 0$
$r(t_1) = |\bar{r}(t_1)|$
$\bar{u}r_1 = \frac{\bar{r}(t_1)}{r(t_1)}$
$c_1 = \frac{\bar{r}(t_1) \cdot \bar{V}(t_1)}{\sqrt{\mu}}$
$c_2 = \frac{r(t_1)\left[\bar{V}(t_1) \cdot \bar{V}(t_1)\right]}{\mu} - 1$
$\alpha = (1 - c_2) / r(t_1)$
$\alpha > 0$
YES
NO
$X_{MAX} = 2\pi$
$X_{MAX} = 7.071068$
LIMIT $X_{MAX} \leq X_{MAX1}$
$T_D > 0$
NO
YES
$X_{MIN} = - X_{MAX}$
$X_{MAX} = 0$
2/2
1/2

Page 1 of 4

Figure B-22.- Flow chart of subroutine CONIC.



Page 2 of 4

Figure B-22.- Continued.

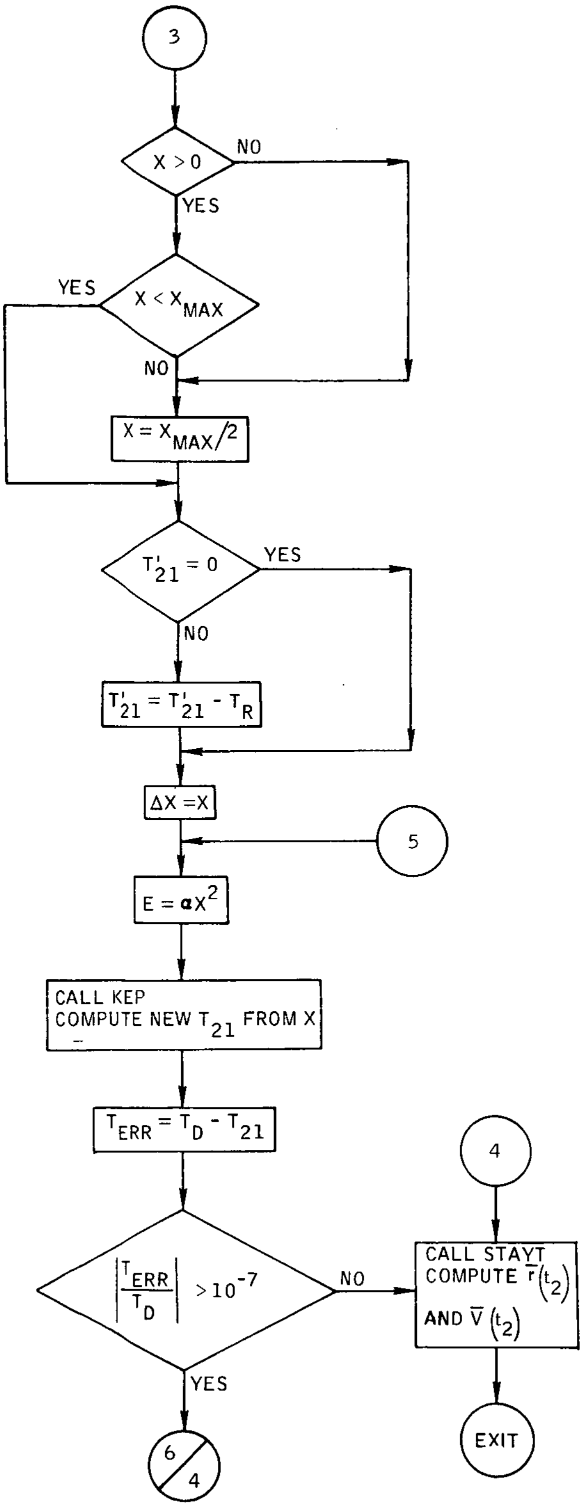
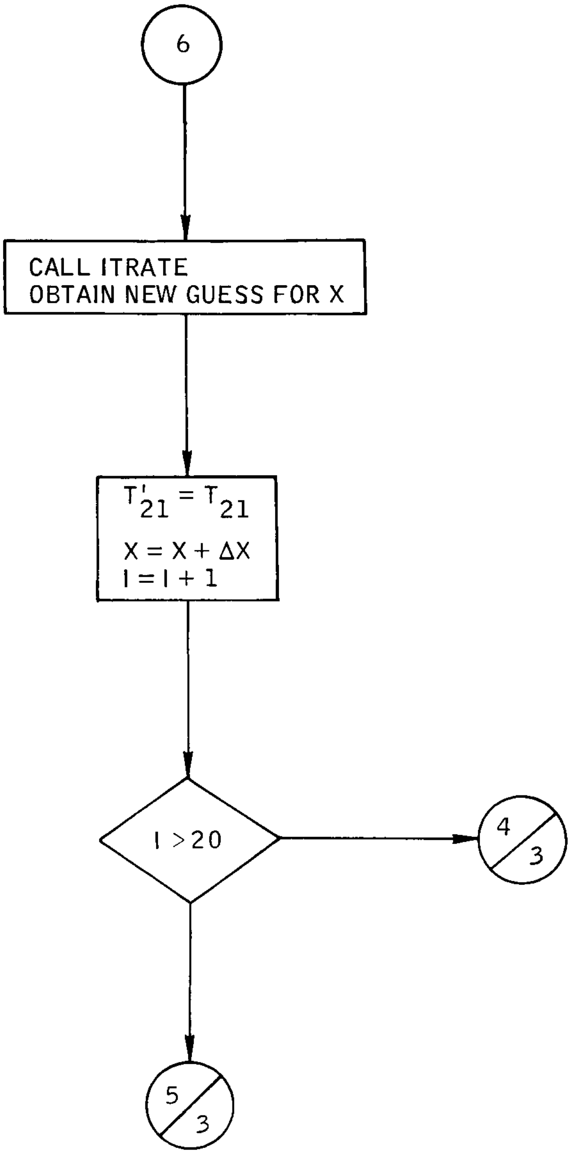
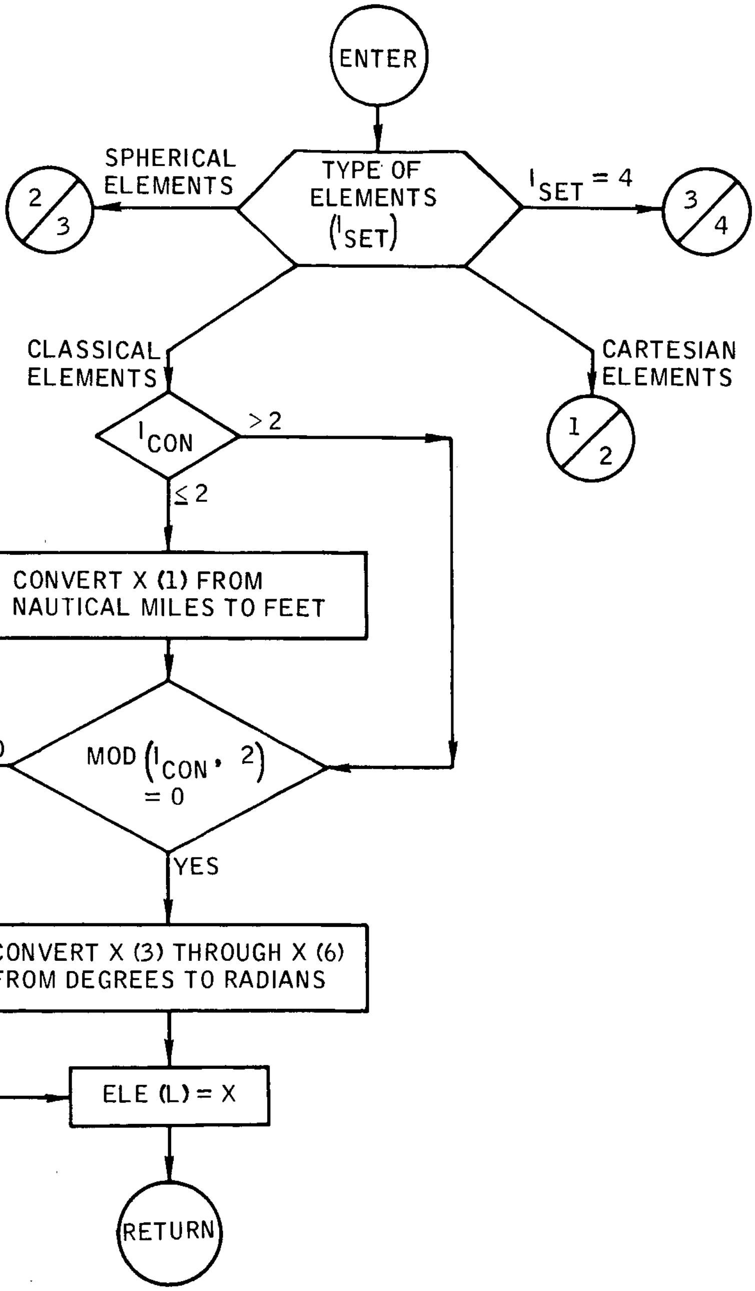


Figure B-22.- Continued.



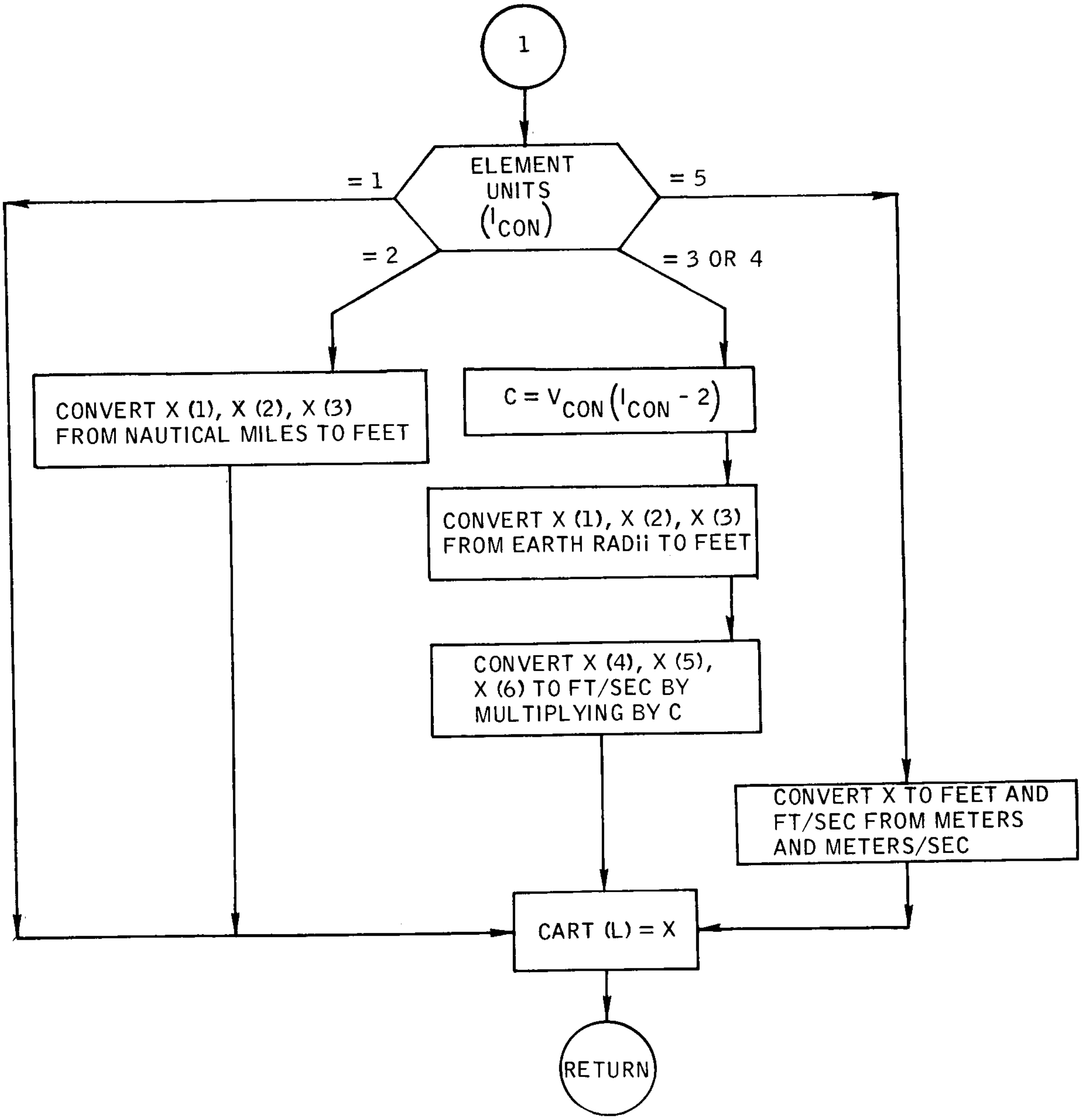
Page 4 of 4

Figure B-22.- Concluded.



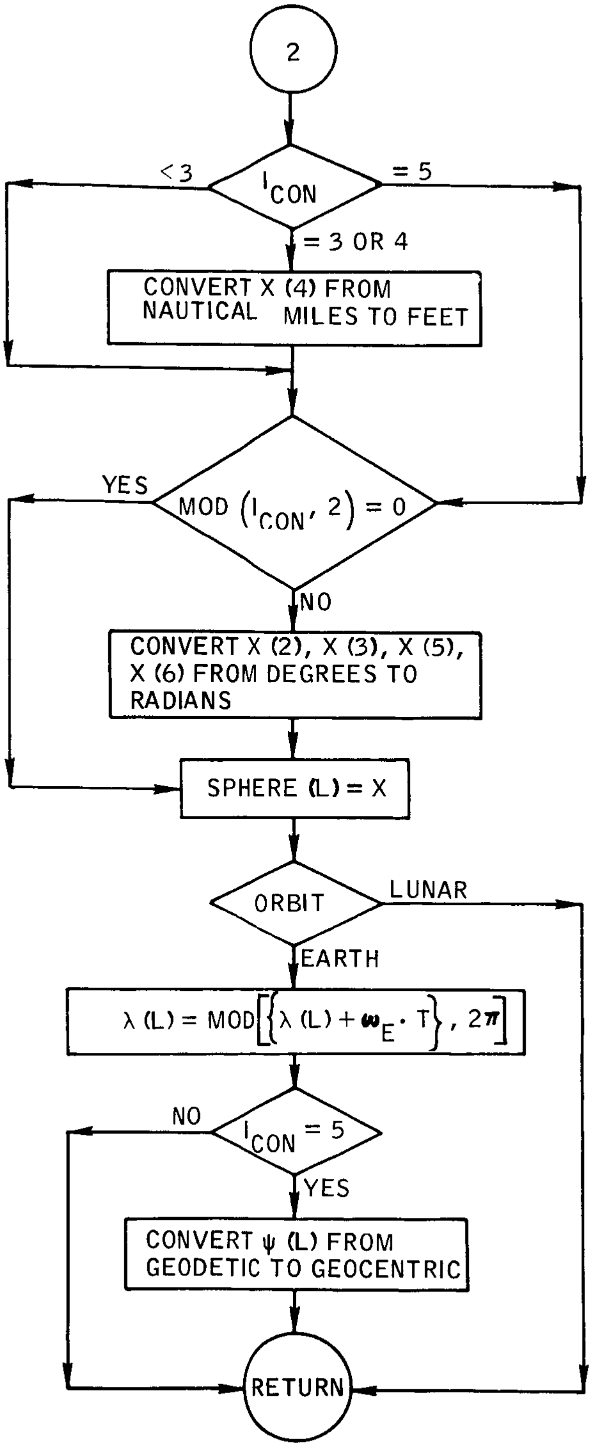
Page 1 of 4

Figure B-23.- Flow chart of subroutine CONVRT.


1
ELEMENT UNITS (I_{CON})
= 1
= 5
= 2
= 3 OR 4
CONVERT X (1), X (2), X (3) FROM NAUTICAL MILES TO FEET
C = $V_{CON}(I_{CON} - 2)$
CONVERT X (1), X (2), X (3) FROM EARTH RADii TO FEET
CONVERT X (4), X (5), X (6) TO FT/SEC BY MULTIPLYING BY C
CONVERT X TO FEET AND FT/SEC FROM METERS AND METERS/SEC
CART (L) = X
RETURN

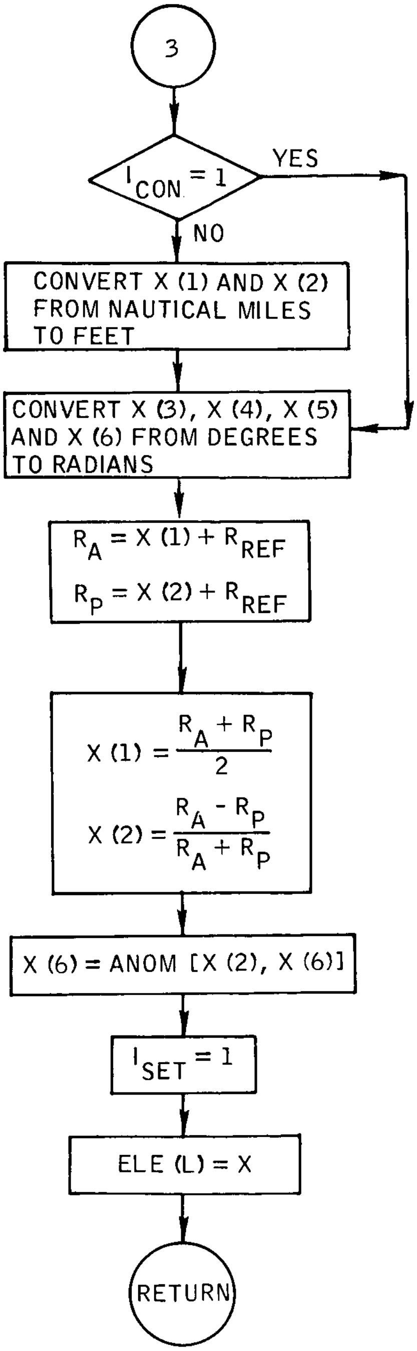
Page 2 of 4

Figure B-23.- Continued.


2
ICON
<3
= 5
= 3 OR 4
CONVERT X (4) FROM NAUTICAL MILES TO FEET
MOD (ICON, 2) = 0
YES
NO
CONVERT X (2), X (3), X (5), X (6) FROM DEGREES TO RADIANS
SPHERE (L) = X
ORBIT
LUNAR
EARTH
λ (L) = MOD[{λ (L) + ωE · T}, 2π]
ICON = 5
NO
YES
CONVERT ψ (L) FROM GEODETIC TO GEOCENTRIC
RETURN

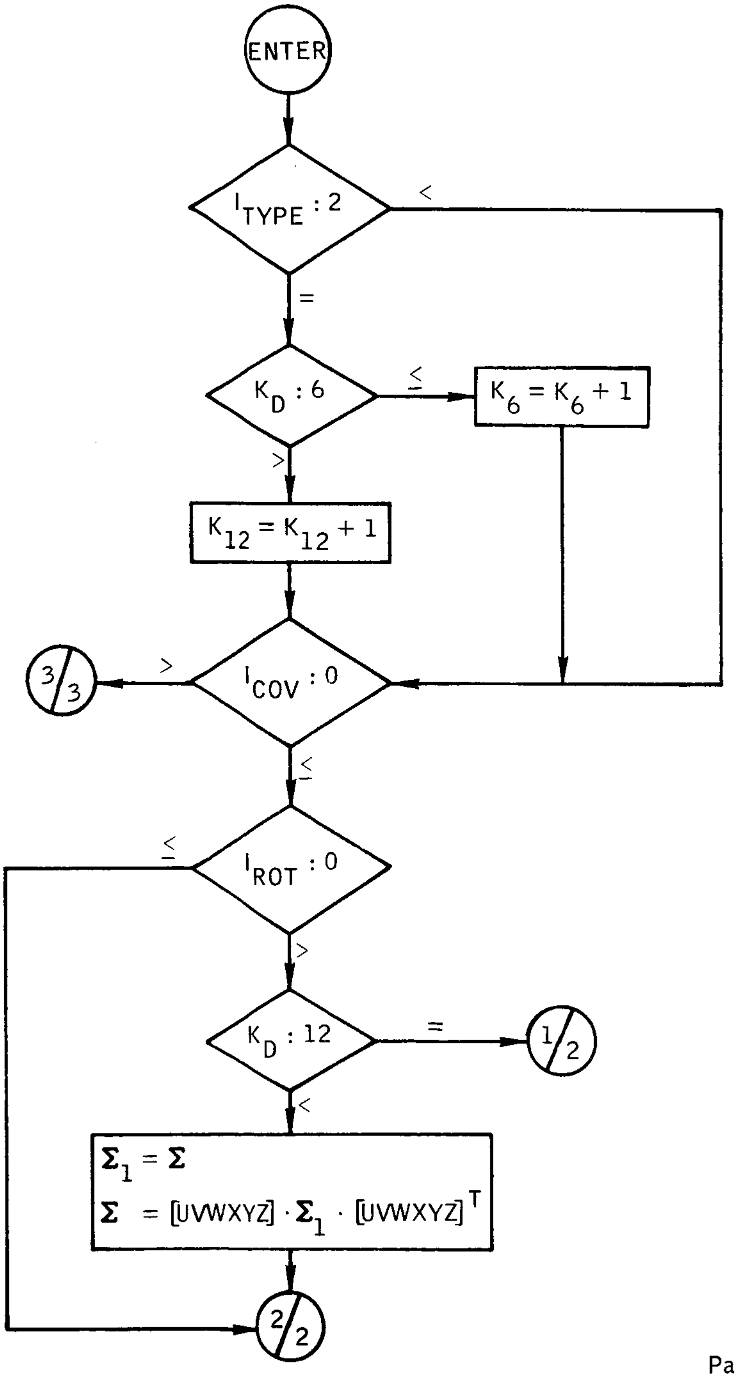
Page 3 of 4

Figure B-23.- Continued.


3
$I_{CON} = 1$
YES
NO
CONVERT X (1) AND X (2) FROM NAUTICAL MILES TO FEET
CONVERT X (3), X (4), X (5) AND X (6) FROM DEGREES TO RADIANS
$R_A = X(1) + R_{REF}$
$R_P = X(2) + R_{REF}$
$X(1) = \frac{R_A + R_P}{2}$
$X(2) = \frac{R_A - R_P}{R_A + R_P}$
X (6) = ANOM [X (2), X (6)]
$I_{SET} = 1$
ELE (L) = X
RETURN

Page 4 of 4

Figure B-23.- Concluded.



Page 1 of 3

Figure B-24.- Flow chart of subroutine COVTAB.

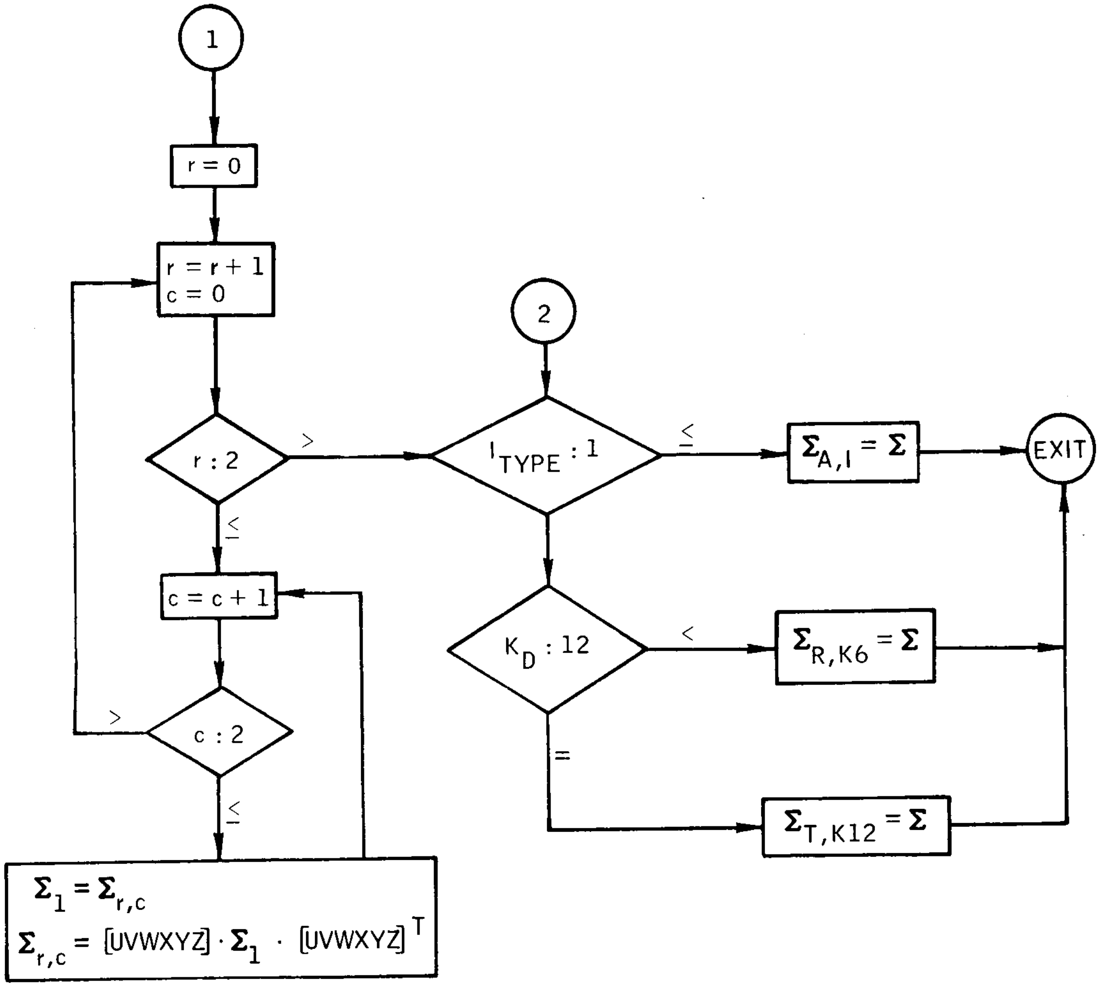


Figure B-24.- Continued.

Page 2 of 3

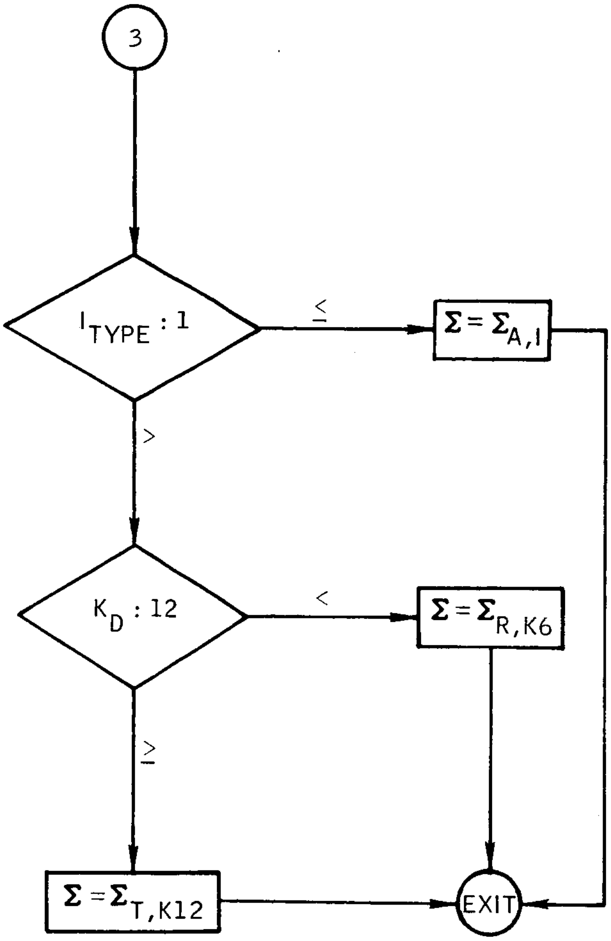


Figure B-24.- Concluded.

Page 3 of 3

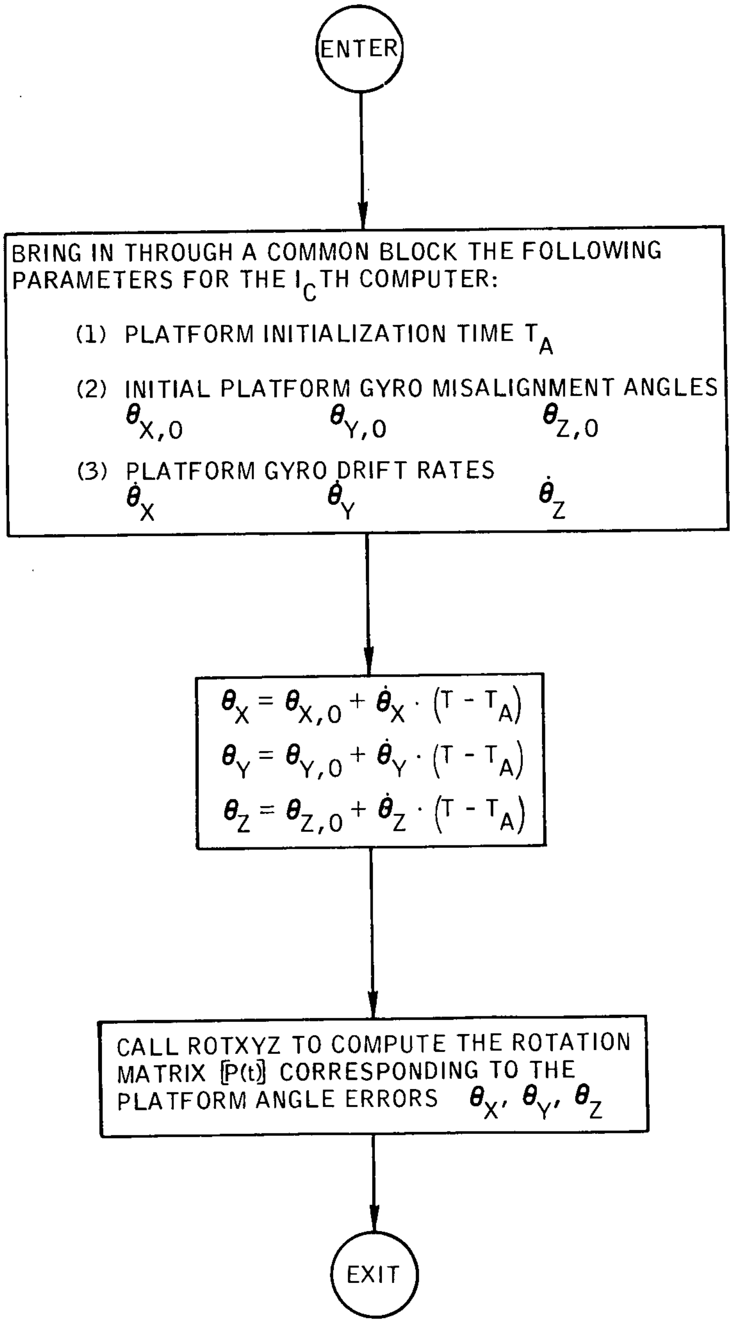
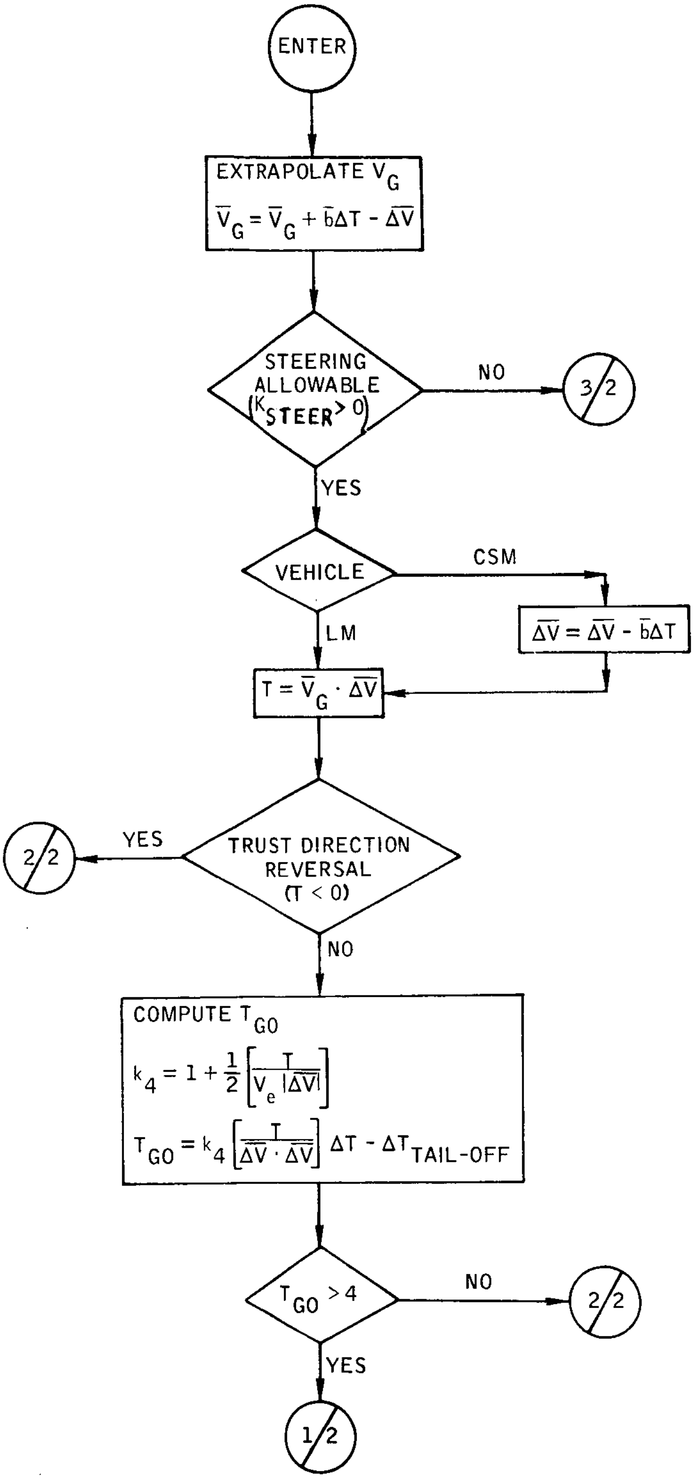

ENTER
BRING IN THROUGH A COMMON BLOCK THE FOLLOWING PARAMETERS FOR THE I_CTH COMPUTER:
(1) PLATFORM INITIALIZATION TIME T_A
(2) INITIAL PLATFORM GYRO MISALIGNMENT ANGLES $\theta_{X,0}$ $\theta_{Y,0}$ $\theta_{Z,0}$
(3) PLATFORM GYRO DRIFT RATES $\dot{\theta}_X$ $\dot{\theta}_Y$ $\dot{\theta}_Z$
$\theta_X = \theta_{X,0} + \dot{\theta}_X \cdot (T - T_A)$
$\theta_Y = \theta_{Y,0} + \dot{\theta}_Y \cdot (T - T_A)$
$\theta_Z = \theta_{Z,0} + \dot{\theta}_Z \cdot (T - T_A)$
CALL ROTXYZ TO COMPUTE THE ROTATION MATRIX $[P(t)]$ CORRESPONDING TO THE PLATFORM ANGLE ERRORS θ_X, θ_Y, θ_Z
EXIT

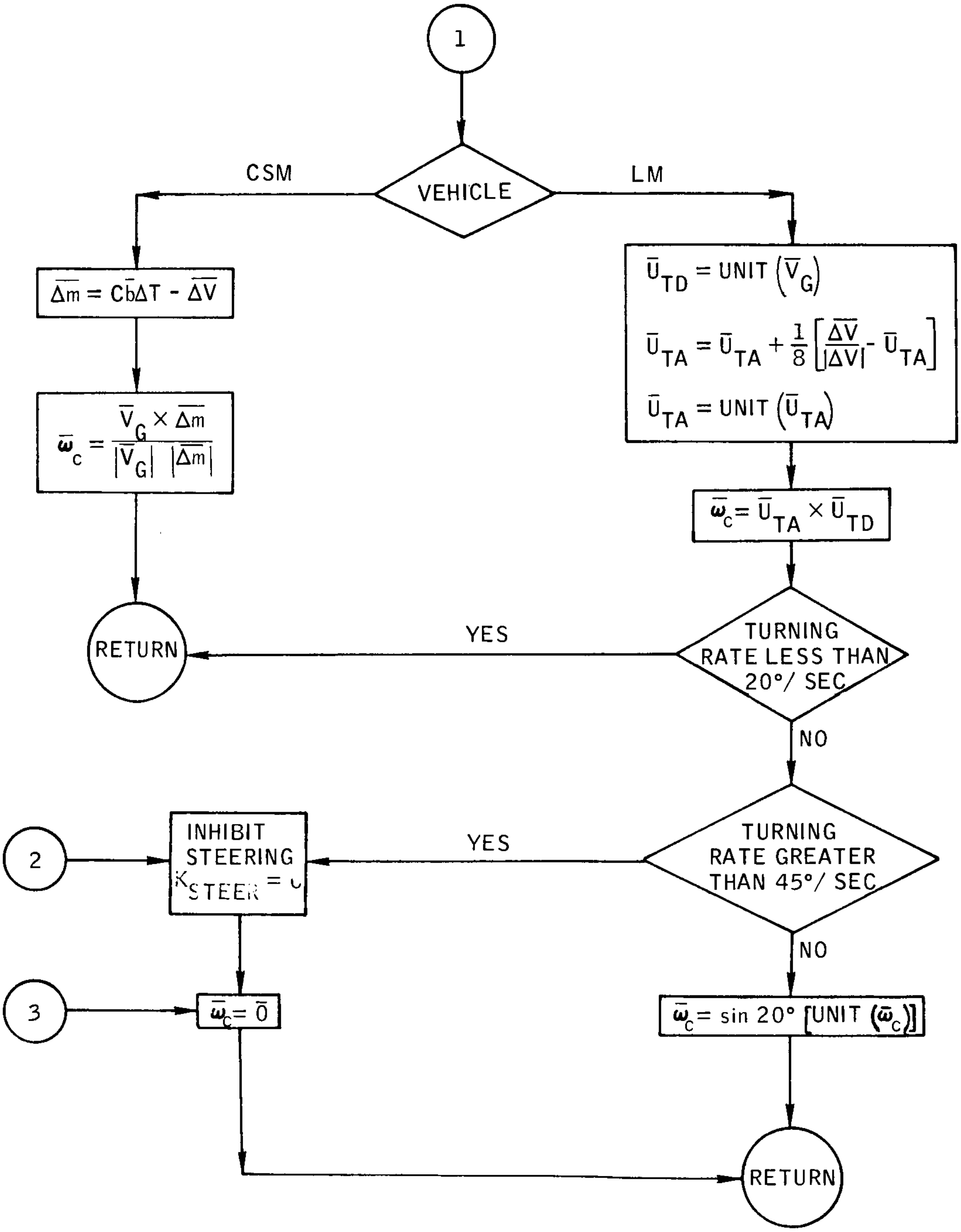
Figure B-25.- Flow chart of subroutine CPEM.

Page 1 of 1


ENTER
EXTRAPOLATE V_G
$\overline{V}_G = \overline{V}_G + \overline{b}\Delta T - \overline{\Delta V}$
STEERING ALLOWABLE ($K_{STEER} > 0$)
NO
3/2
YES
VEHICLE
CSM
$\overline{\Delta V} = \overline{\Delta V} - \overline{b}\Delta T$
LM
$T = \overline{V}_G \cdot \overline{\Delta V}$
TRUST DIRECTION REVERSAL ($T < 0$)
YES
2/2
NO
COMPUTE T_{GO}
$k_4 = 1 + \frac{1}{2}\left[\frac{T}{V_e |\overline{\Delta V}|}\right]$
$T_{GO} = k_4 \left[\frac{T}{\overline{\Delta V} \cdot \overline{\Delta V}}\right] \Delta T - \Delta T_{TAIL-OFF}$
$T_{GO} > 4$
NO
2/2
YES
1/2

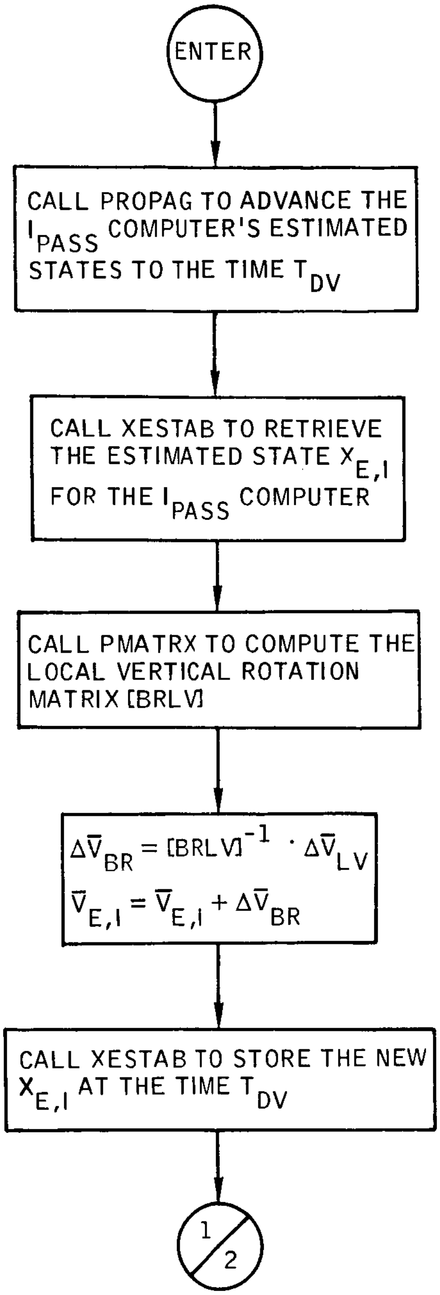
Page 1 of 2

Figure B-26.- Flow chart of subroutine CROSP.


1
VEHICLE
CSM
LM
$\overline{\Delta m} = C\bar{b}\Delta T - \overline{\Delta V}$
$\bar{\omega}_c = \frac{\bar{V}_G \times \overline{\Delta m}}{|\bar{V}_G|\ |\overline{\Delta m}|}$
$\bar{U}_{TD} = \text{UNIT}\left(\bar{V}_G\right)$
$\bar{U}_{TA} = \bar{U}_{TA} + \frac{1}{8}\left[\frac{\overline{\Delta V}}{|\overline{\Delta V}|} - \bar{U}_{TA}\right]$
$\bar{U}_{TA} = \text{UNIT}\left(\bar{U}_{TA}\right)$
$\bar{\omega}_c = \bar{U}_{TA} \times \bar{U}_{TD}$
TURNING RATE LESS THAN 20°/ SEC
YES
NO
RETURN
TURNING RATE GREATER THAN 45°/ SEC
YES
NO
2
INHIBIT STEERING $K_{STEER} = 0$
3
$\bar{\omega}_c = \bar{0}$
$\bar{\omega}_c = \sin 20° \left[\text{UNIT}\left(\bar{\omega}_c\right)\right]$
RETURN

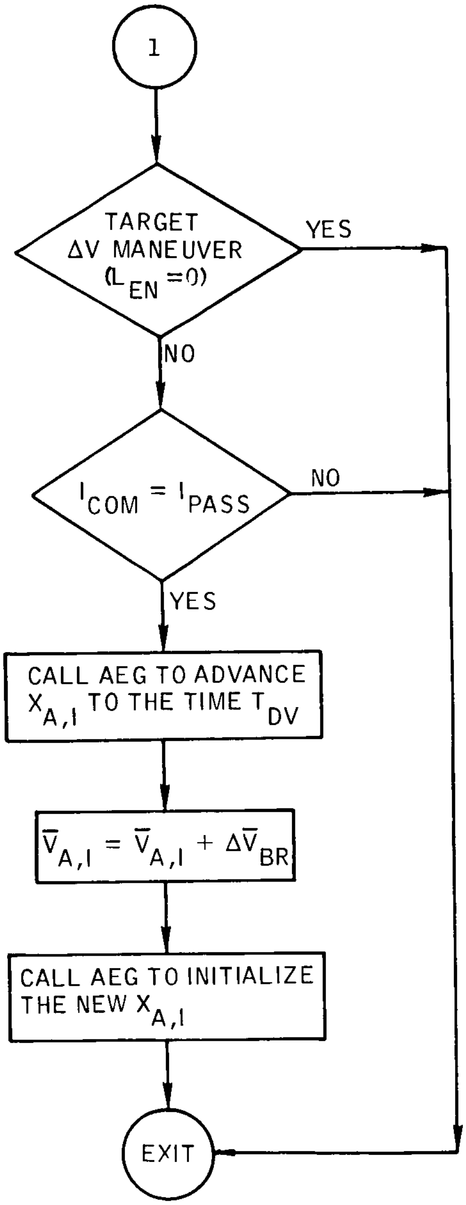
Page 2 of 2

Figure B-26.- Concluded.


ENTER
CALL PROPAG TO ADVANCE THE I_{PASS} COMPUTER'S ESTIMATED STATES TO THE TIME T_{DV}
CALL XESTAB TO RETRIEVE THE ESTIMATED STATE $X_{E,I}$ FOR THE I_{PASS} COMPUTER
CALL PMATRX TO COMPUTE THE LOCAL VERTICAL ROTATION MATRIX [BRLV]
$\Delta\bar{V}_{BR} = [BRLV]^{-1} \cdot \Delta\bar{V}_{LV}$
$\bar{V}_{E,I} = \bar{V}_{E,I} + \Delta\bar{V}_{BR}$
CALL XESTAB TO STORE THE NEW $X_{E,I}$ AT THE TIME T_{DV}
1/2

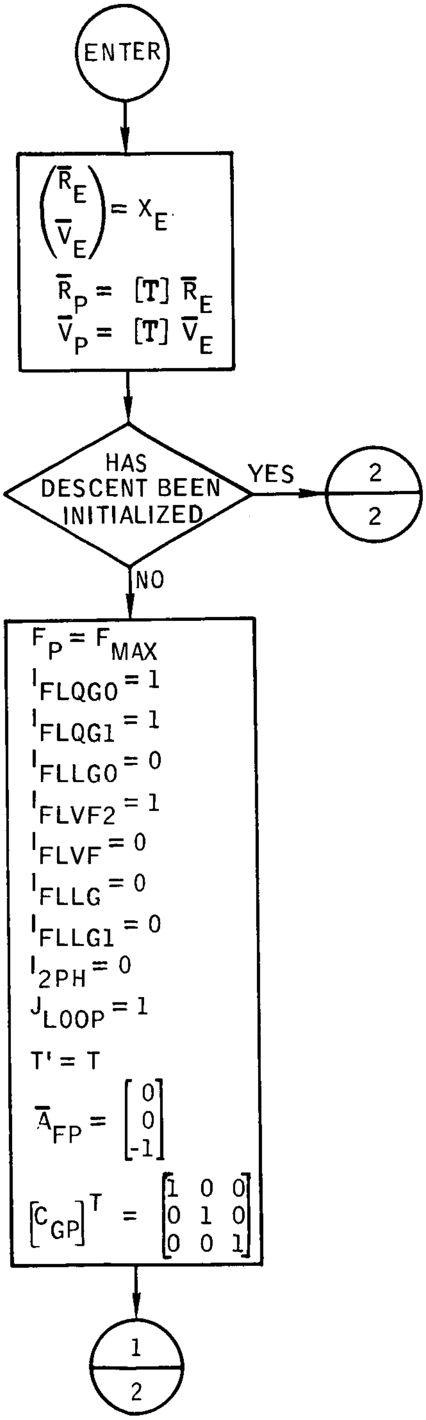
Page 1 of 2

Figure B-27.- Flow chart of subroutine DELTAV.



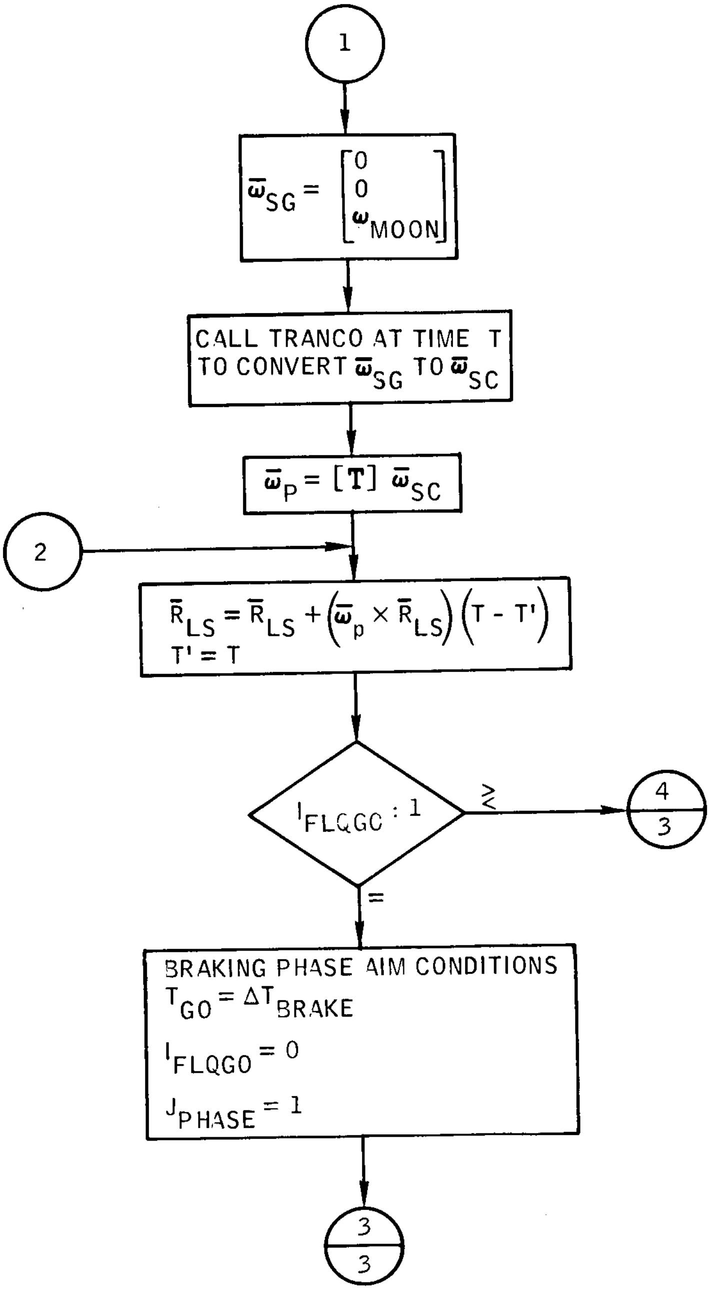
Page 2 of 2

Figure B-27.- Concluded.


ENTER
$\begin{pmatrix}\bar{R}_E\\ \bar{V}_E\end{pmatrix} = X_E$
$\bar{R}_P = [T]\,\bar{R}_E$
$\bar{V}_P = [T]\,\bar{V}_E$
HAS DESCENT BEEN INITIALIZED
YES
2
2
NO
$F_P = F_{MAX}$
$I_{FLQGO} = 1$
$I_{FLQG1} = 1$
$I_{FLLGO} = 0$
$I_{FLVF2} = 1$
$I_{FLVF} = 0$
$I_{FLLG} = 0$
$I_{FLLG1} = 0$
$I_{2PH} = 0$
$J_{LOOP} = 1$
$T' = T$
$\bar{A}_{FP} = \begin{bmatrix}0\\0\\-1\end{bmatrix}$
$\left[C_{GP}\right]^T = \begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$
1
2

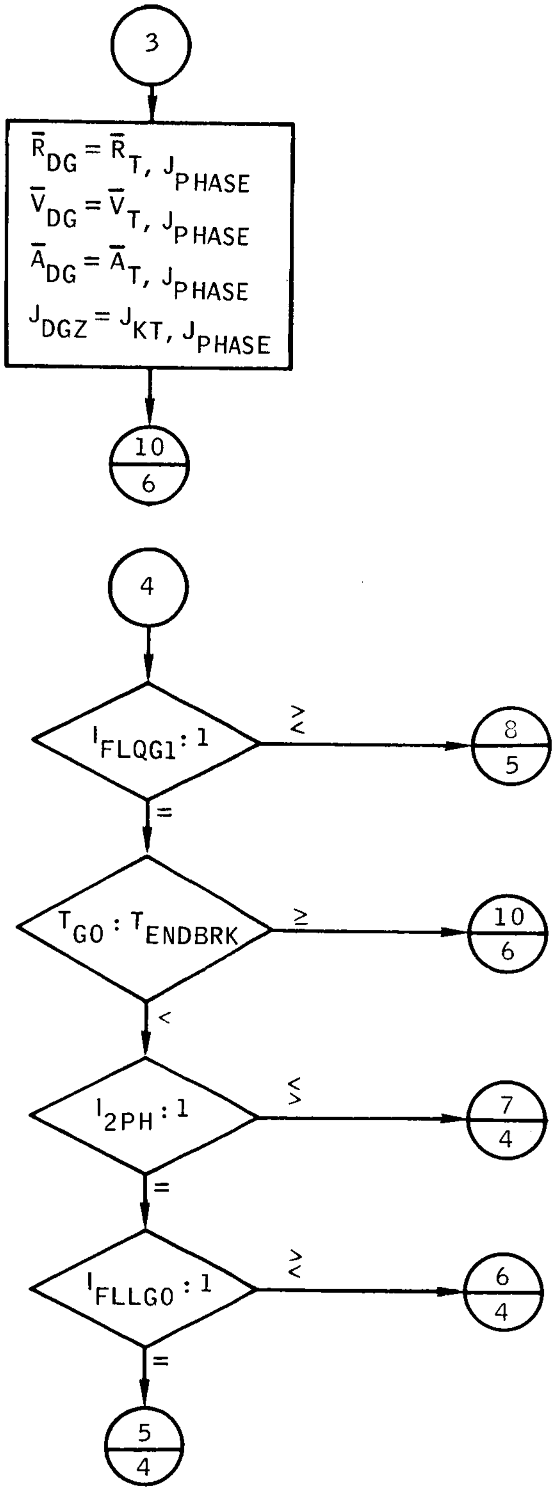
Page 1 of 14

Figure B-28.- Flow chart of subroutine DESCENT.


1
$\bar{\omega}_{SG} = \begin{bmatrix} 0 \\ 0 \\ \omega_{MOON} \end{bmatrix}$
CALL TRANCO AT TIME T TO CONVERT $\bar{\omega}_{SG}$ TO $\bar{\omega}_{SC}$
$\bar{\omega}_P = [T]\, \bar{\omega}_{SC}$
2
$\bar{R}_{LS} = \bar{R}_{LS} + \left(\bar{\omega}_p \times \bar{R}_{LS}\right)\left(T - T'\right)$
$T' = T$
$I_{FLQGO} : 1$
$\gtrless$
4/3
=
BRAKING PHASE AIM CONDITIONS
$T_{GO} = \Delta T_{BRAKE}$
$I_{FLQGO} = 0$
$J_{PHASE} = 1$
3/3

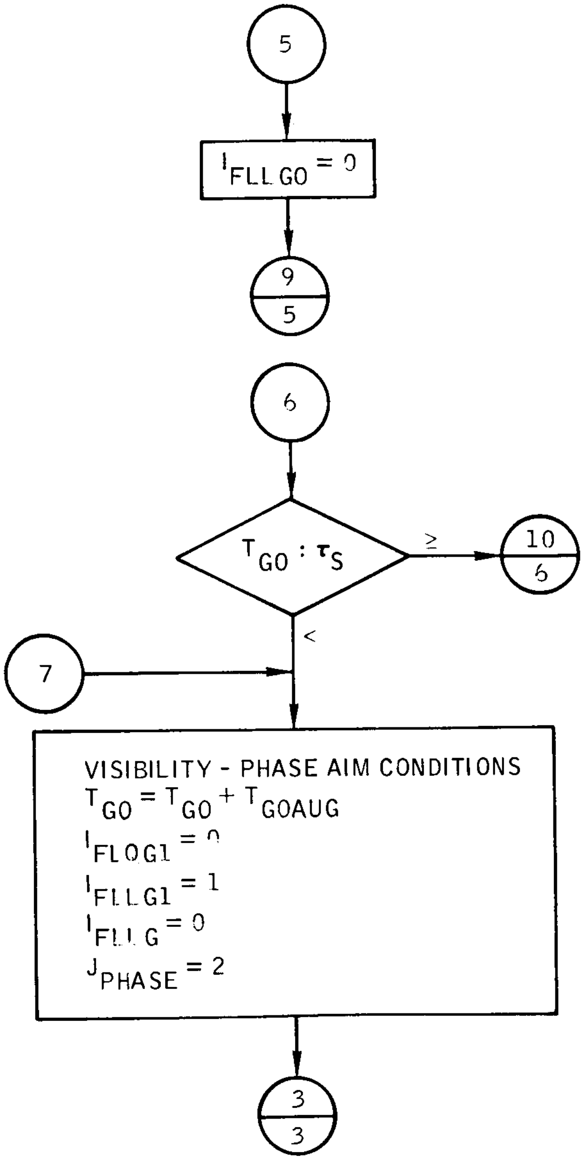
Page 2 of 14

Figure B-28.- Continued.


3
$\bar{R}_{DG} = \bar{R}_{T, J_{PHASE}}$
$\bar{V}_{DG} = \bar{V}_{T, J_{PHASE}}$
$\bar{A}_{DG} = \bar{A}_{T, J_{PHASE}}$
$J_{DGZ} = J_{KT, J_{PHASE}}$
10
6
4
$I_{FLQG1} : 1$
$\gtrless$
8
5
=
$T_{GO} : T_{ENDBRK}$
≥
10
6
<
$I_{2PH} : 1$
$\lessgtr$
7
4
=
$I_{FLLGO} : 1$
$\gtrless$
6
4
=
5
4

Page 3 of 14

Figure B-28.- Continued.



Page 4 of 14

Figure B-28.- Continued.

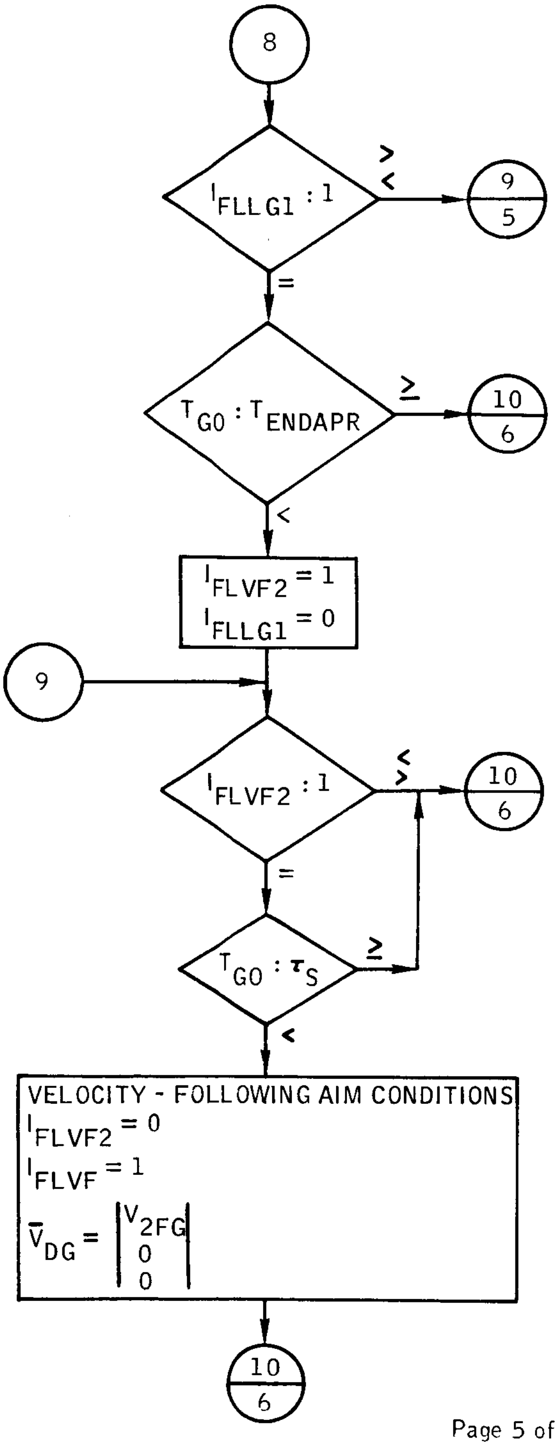
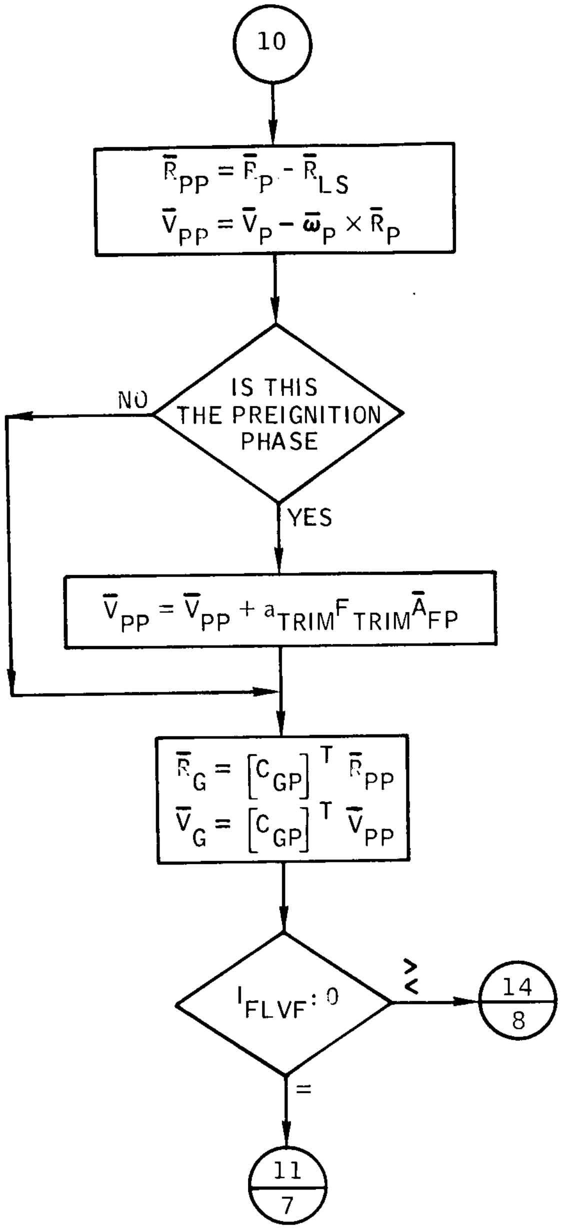
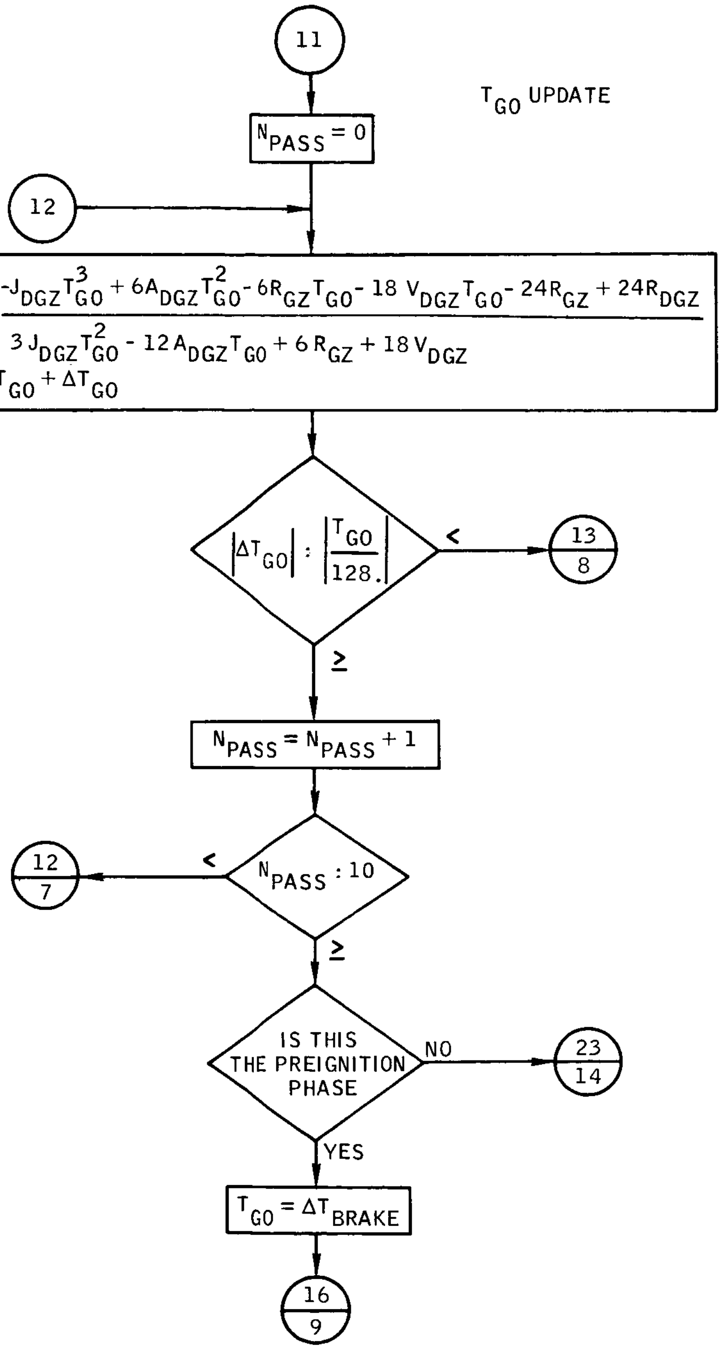


Figure B-28.- Continued.


10
$\bar{R}_{PP} = \bar{R}_P - \bar{R}_{LS}$
$\bar{V}_{PP} = \bar{V}_P - \bar{\omega}_P \times \bar{R}_P$
IS THIS THE PREIGNITION PHASE
NO
YES
$\bar{V}_{PP} = \bar{V}_{PP} + a_{TRIM} F_{TRIM} \bar{A}_{FP}$
$\bar{R}_G = [C_{GP}]^T \bar{R}_{PP}$
$\bar{V}_G = [C_{GP}]^T \bar{V}_{PP}$
$I_{FLVF} : 0$
$>$
$<$
$=$
14/8
11/7

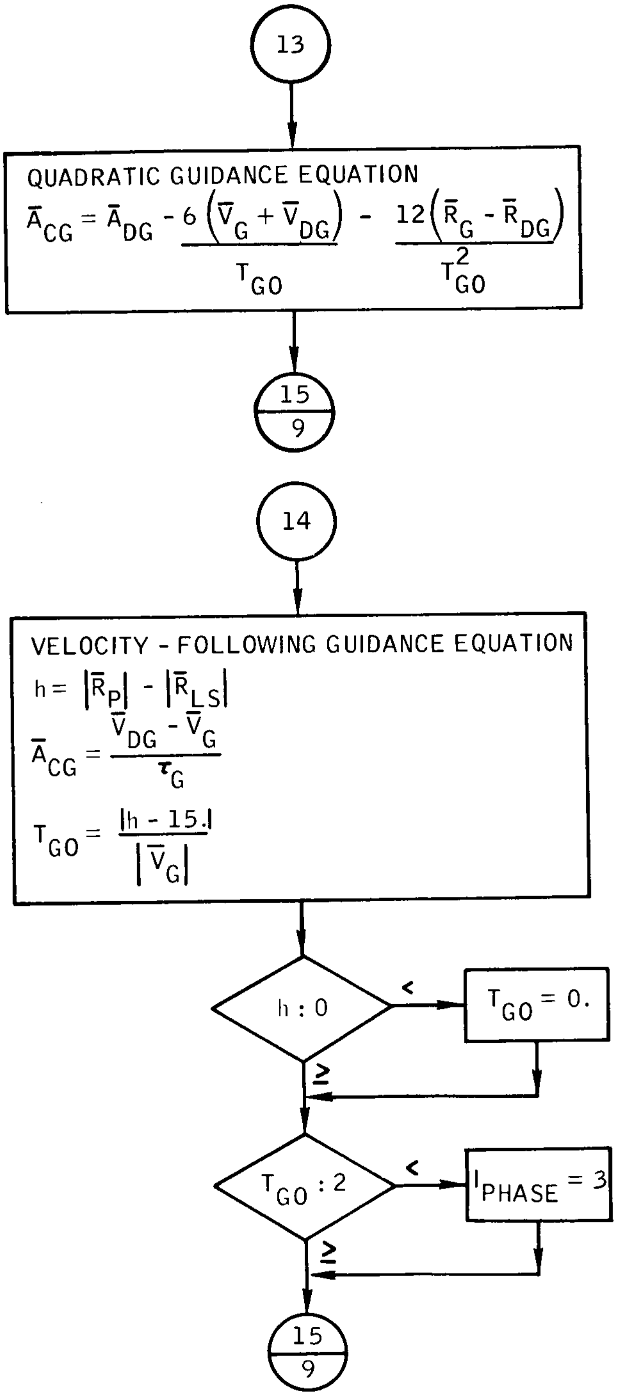
Page 6 of 14

Figure B-28.- Continued.


11
T_{GO} UPDATE
$N_{PASS} = 0$
12
$\Delta T_{GO} = \frac{-J_{DGZ}T_{GO}^3 + 6A_{DGZ}T_{GO}^2 - 6R_{GZ}T_{GO} - 18\,V_{DGZ}T_{GO} - 24R_{GZ} + 24R_{DGZ}}{3\,J_{DGZ}T_{GO}^2 - 12\,A_{DGZ}T_{GO} + 6\,R_{GZ} + 18\,V_{DGZ}}$
$T_{GO} = T_{GO} + \Delta T_{GO}$
$\left|\Delta T_{GO}\right| : \left|\frac{T_{GO}}{128.}\right|$
<
13/8
≥
$N_{PASS} = N_{PASS} + 1$
$N_{PASS} : 10$
<
12/7
≥
IS THIS THE PREIGNITION PHASE
NO
23/14
YES
$T_{GO} = \Delta T_{BRAKE}$
16/9

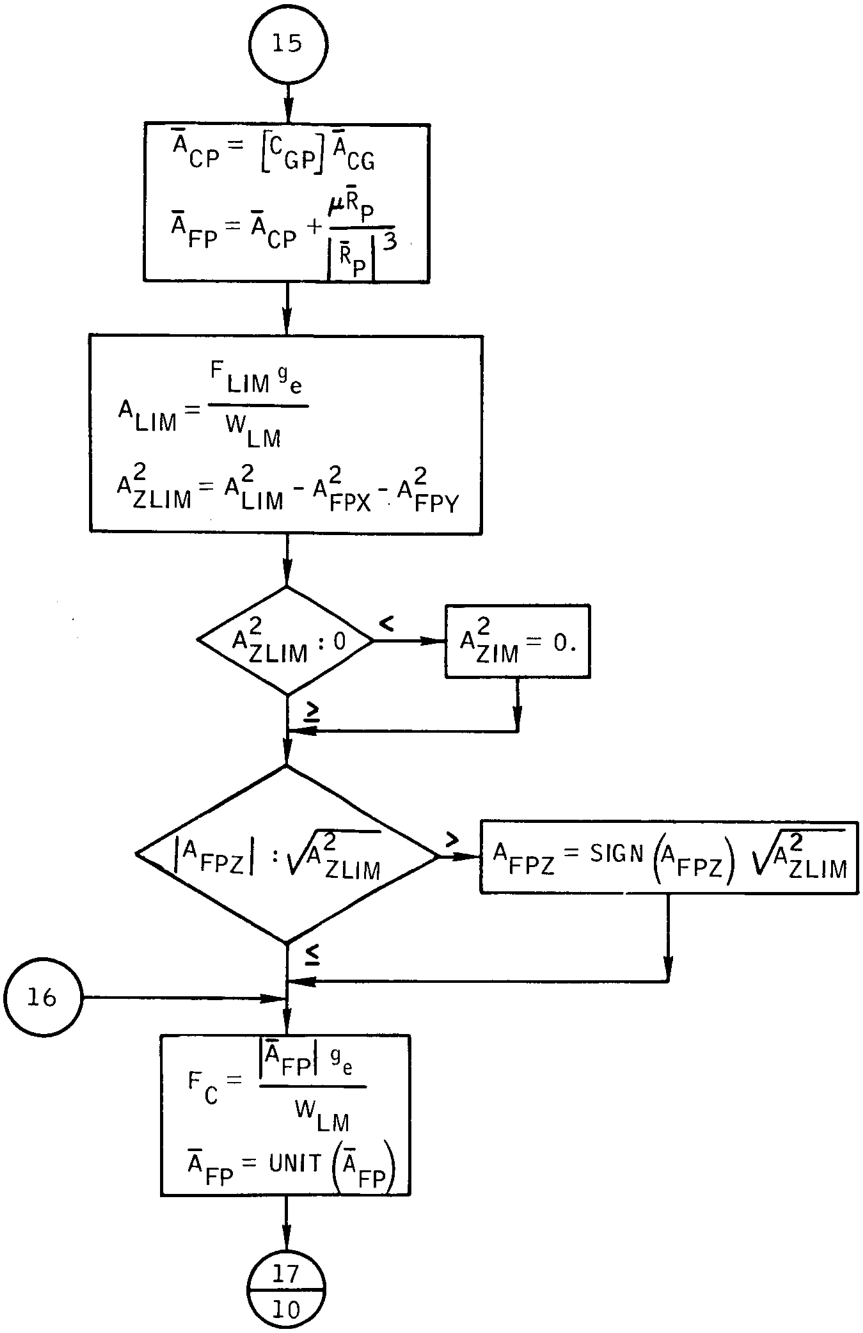
Page 7 of 14

Figure B-28.- Continued.


13
QUADRATIC GUIDANCE EQUATION
$\bar{A}_{CG} = \bar{A}_{DG} - \frac{6\left(\bar{V}_G + \bar{V}_{DG}\right)}{T_{GO}} - \frac{12\left(\bar{R}_G - \bar{R}_{DG}\right)}{T_{GO}^2}$
15/9
14
VELOCITY - FOLLOWING GUIDANCE EQUATION
$h = \left|\bar{R}_P\right| - \left|\bar{R}_{LS}\right|$
$\bar{A}_{CG} = \frac{\bar{V}_{DG} - \bar{V}_G}{\tau_G}$
$T_{GO} = \frac{|h - 15.|}{\left|\bar{V}_G\right|}$
h : 0
<
$T_{GO} = 0.$
≥
$T_{GO} : 2$
<
$I_{PHASE} = 3$
≥
15/9

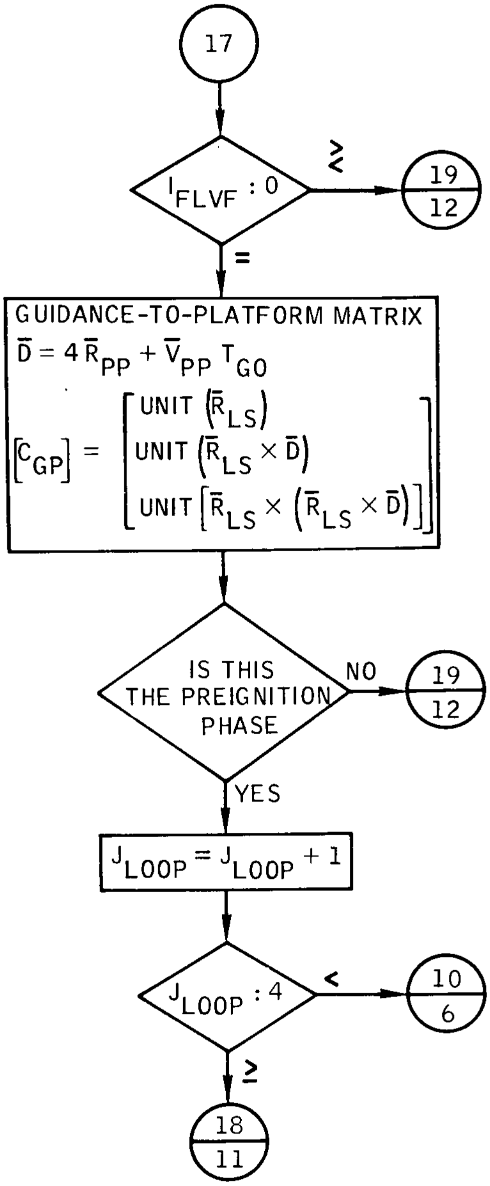
Page 8 of 14

Figure B-28.- Continued.


15
$\bar{A}_{CP} = [C_{GP}]\bar{A}_{CG}$
$\bar{A}_{FP} = \bar{A}_{CP} + \frac{\mu\bar{R}_P}{|\bar{R}_P|^3}$
$A_{LIM} = \frac{F_{LIM}\, g_e}{W_{LM}}$
$A^2_{ZLIM} = A^2_{LIM} - A^2_{FPX} - A^2_{FPY}$
$A^2_{ZLIM} : 0$
<
$A^2_{ZIM} = 0.$
≥
$|A_{FPZ}| : \sqrt{A^2_{ZLIM}}$
>
$A_{FPZ} = \text{SIGN}\,(A_{FPZ})\, \sqrt{A^2_{ZLIM}}$
≤
16
$F_C = \frac{|\bar{A}_{FP}|\, g_e}{W_{LM}}$
$\bar{A}_{FP} = \text{UNIT}\,(\bar{A}_{FP})$
17
10

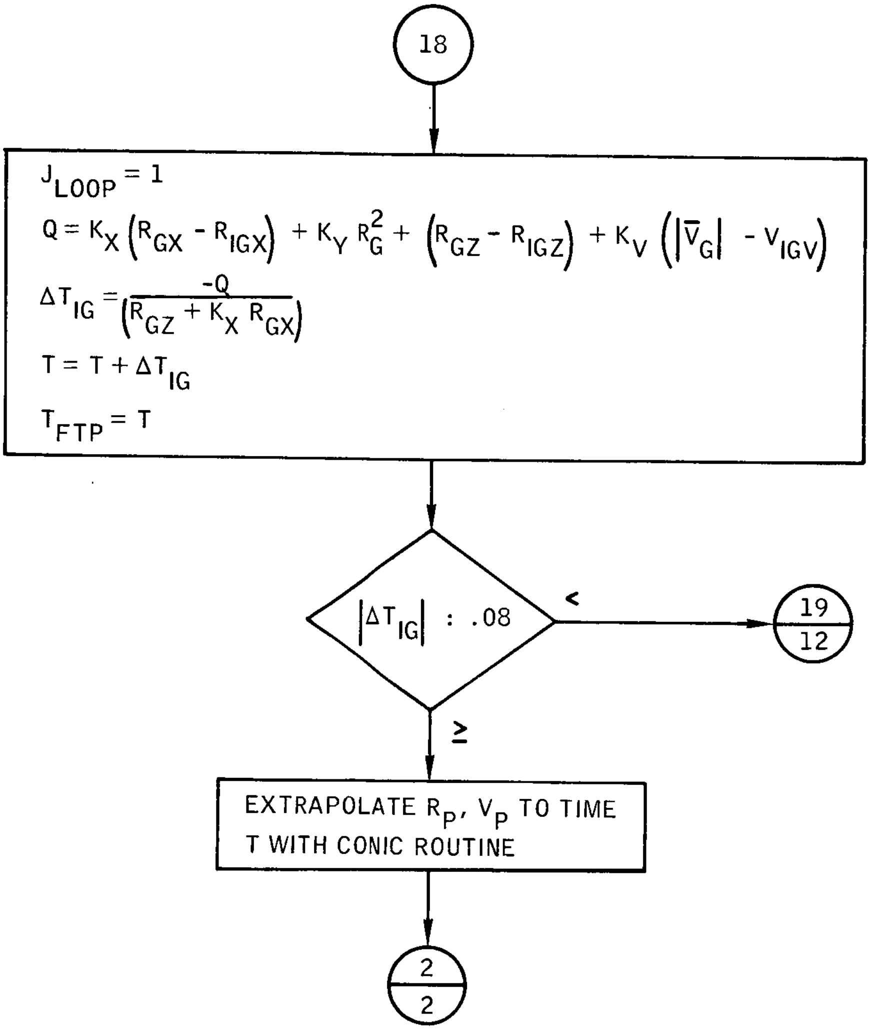
Page 9 of 14

Figure B-28.- Continued.


17
$I_{FLVF} : 0$
$\gtrless$
19/12
=
GUIDANCE-TO-PLATFORM MATRIX
$\bar{D} = 4\bar{R}_{PP} + \bar{V}_{PP} T_{GO}$
$[C_{GP}] = \begin{bmatrix} \text{UNIT}(\bar{R}_{LS}) \\ \text{UNIT}(\bar{R}_{LS} \times \bar{D}) \\ \text{UNIT}[\bar{R}_{LS} \times (\bar{R}_{LS} \times \bar{D})] \end{bmatrix}$
IS THIS THE PREIGNITION PHASE
NO
19/12
YES
$J_{LOOP} = J_{LOOP} + 1$
$J_{LOOP} : 4$
<
10/6
≥
18/11

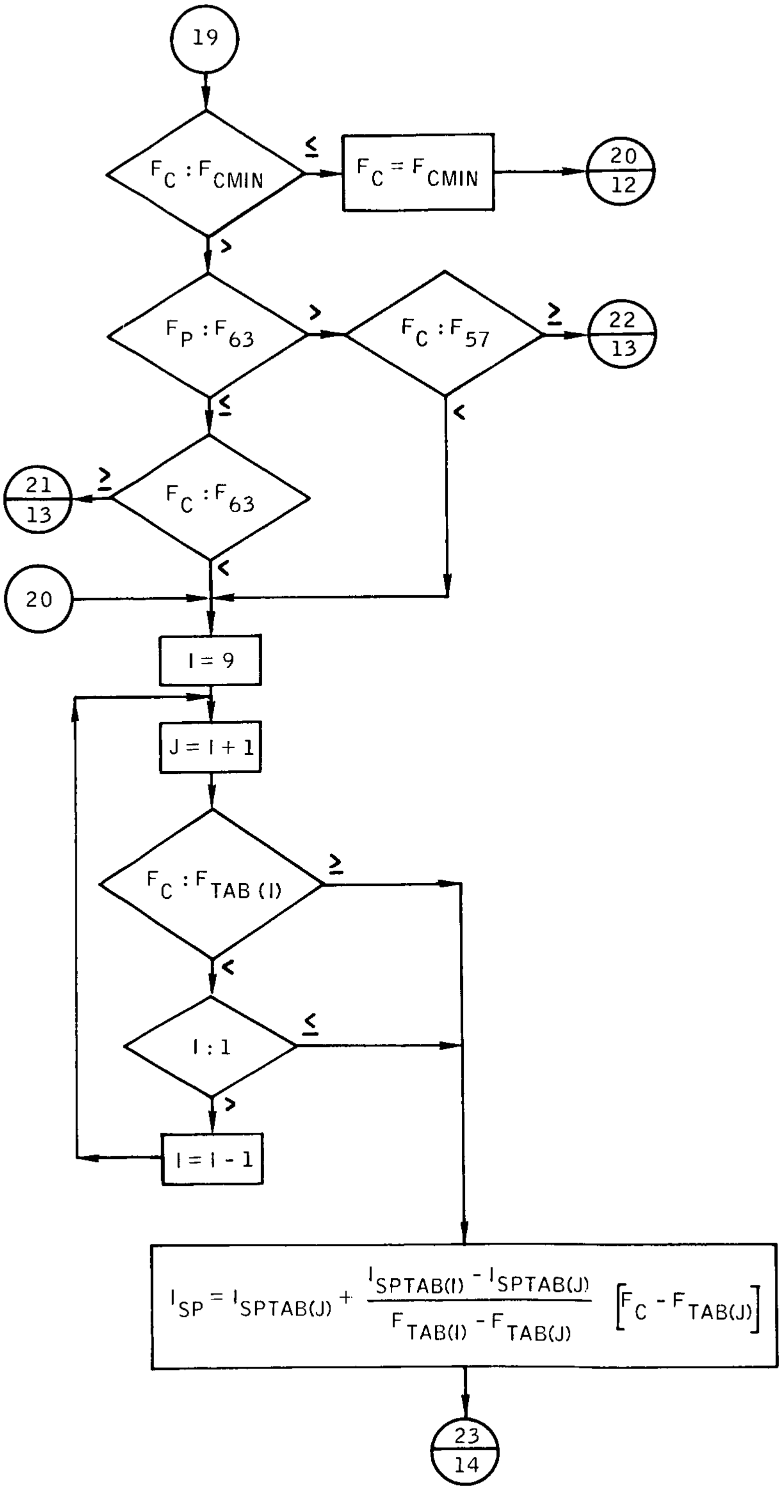
Page 10 of 14

Figure B-28.- Continued.


18
$J_{LOOP} = 1$
$Q = K_X \left(R_{GX} - R_{IGX}\right) + K_Y R_G^2 + \left(R_{GZ} - R_{IGZ}\right) + K_V \left(\left|\bar{V}_G\right| - V_{IGV}\right)$
$\Delta T_{IG} = \frac{-Q}{\left(R_{GZ} + K_X R_{GX}\right)}$
$T = T + \Delta T_{IG}$
$T_{FTP} = T$
$\left|\Delta T_{IG}\right| : .08$
<
19
12
≥
EXTRAPOLATE R_P, V_P TO TIME T WITH CONIC ROUTINE
2
2

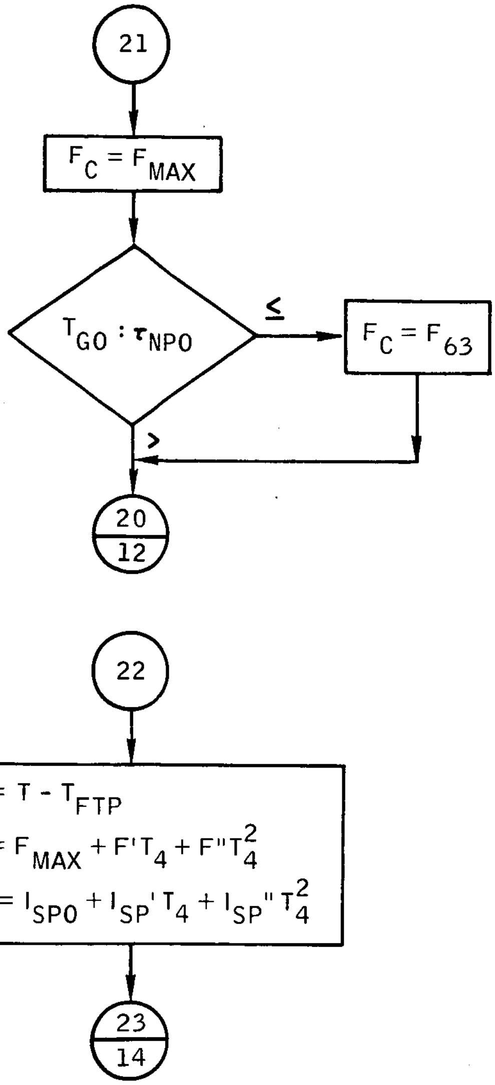
Page 11 of 14

Figure B-28.- Continued.



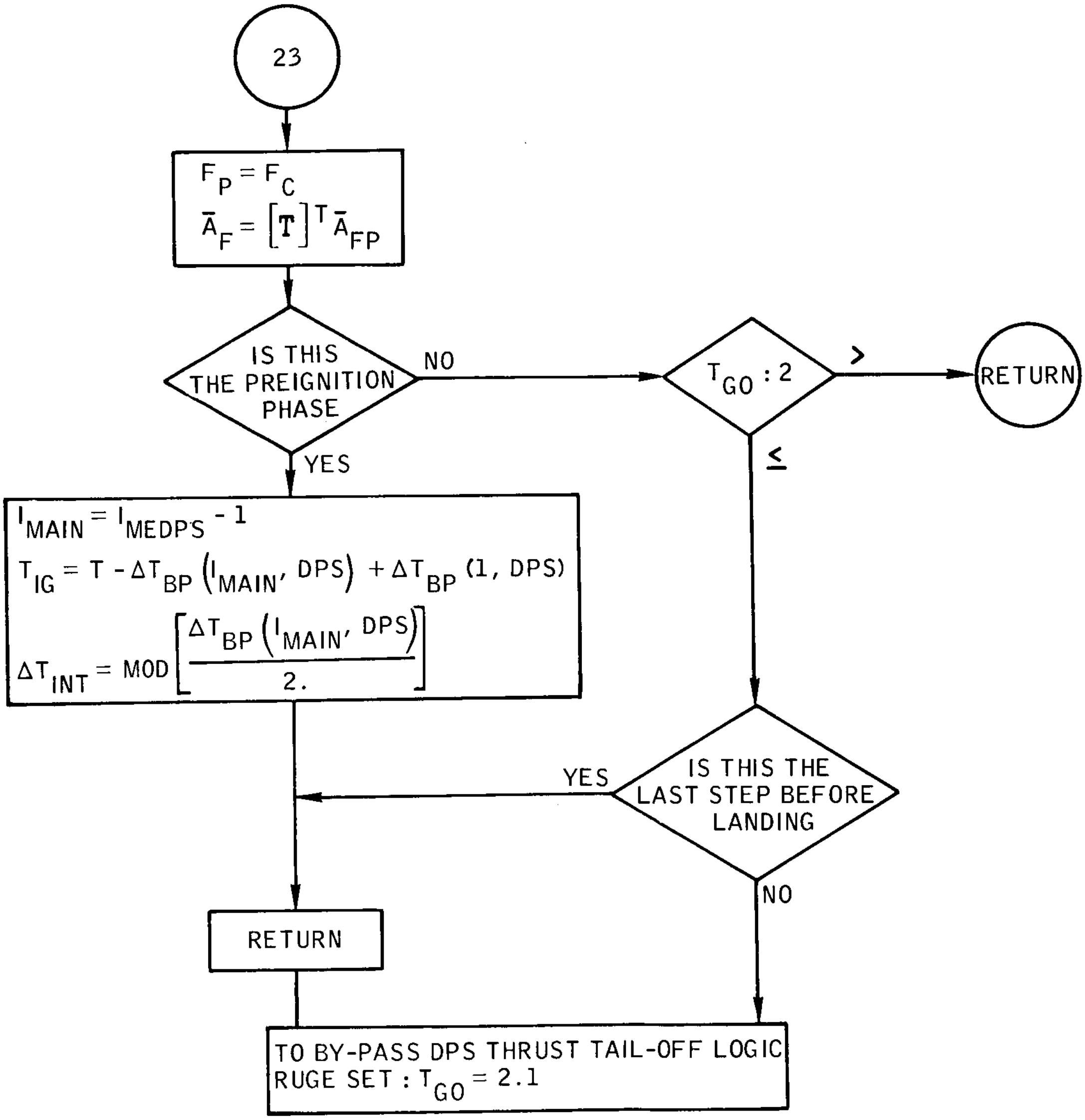
Page 12 of 14

Figure B-28.- Continued.



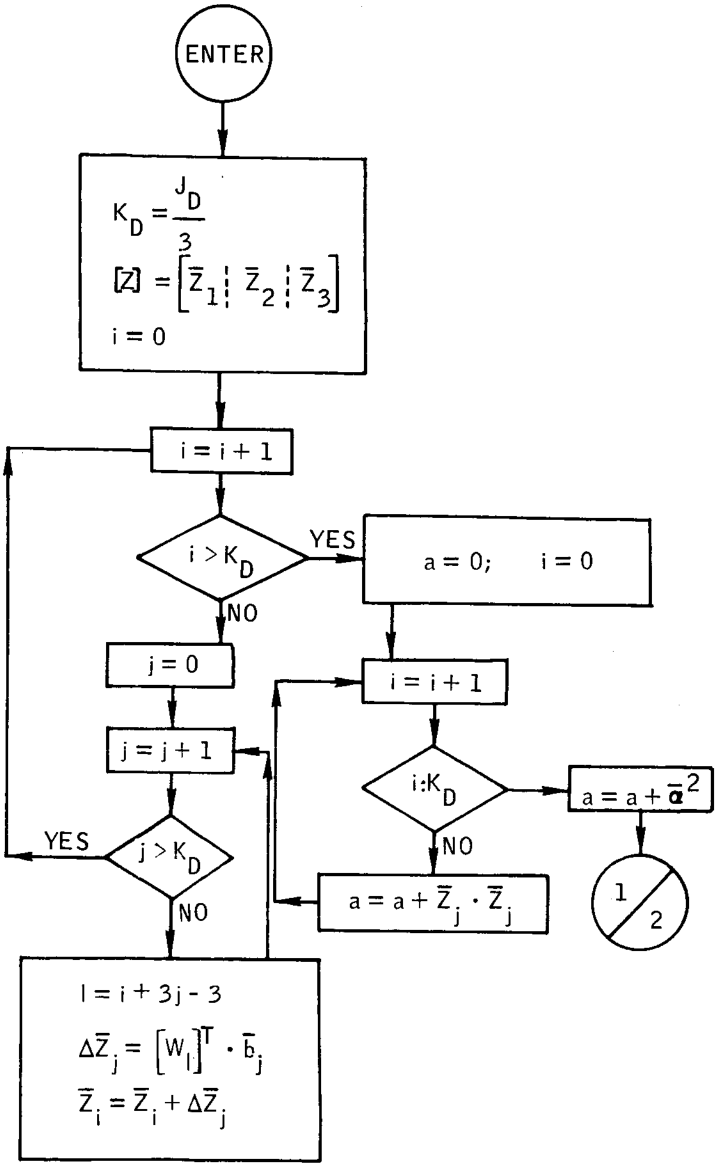
Page 13 of 14

Figure B-28.- Continued.


23
$F_P = F_C$
$\bar{A}_F = [T]^T \bar{A}_{FP}$
IS THIS THE PREIGNITION PHASE
NO
$T_{GO} : 2$
>
RETURN
≤
YES
$I_{MAIN} = I_{MEDPS} - 1$
$T_{IG} = T - \Delta T_{BP}(I_{MAIN}, DPS) + \Delta T_{BP}(1, DPS)$
$\Delta T_{INT} = MOD\left[\frac{\Delta T_{BP}(I_{MAIN}, DPS)}{2.}\right]$
YES
IS THIS THE LAST STEP BEFORE LANDING
NO
RETURN
TO BY-PASS DPS THRUST TAIL-OFF LOGIC
RUGE SET : $T_{GO} = 2.1$

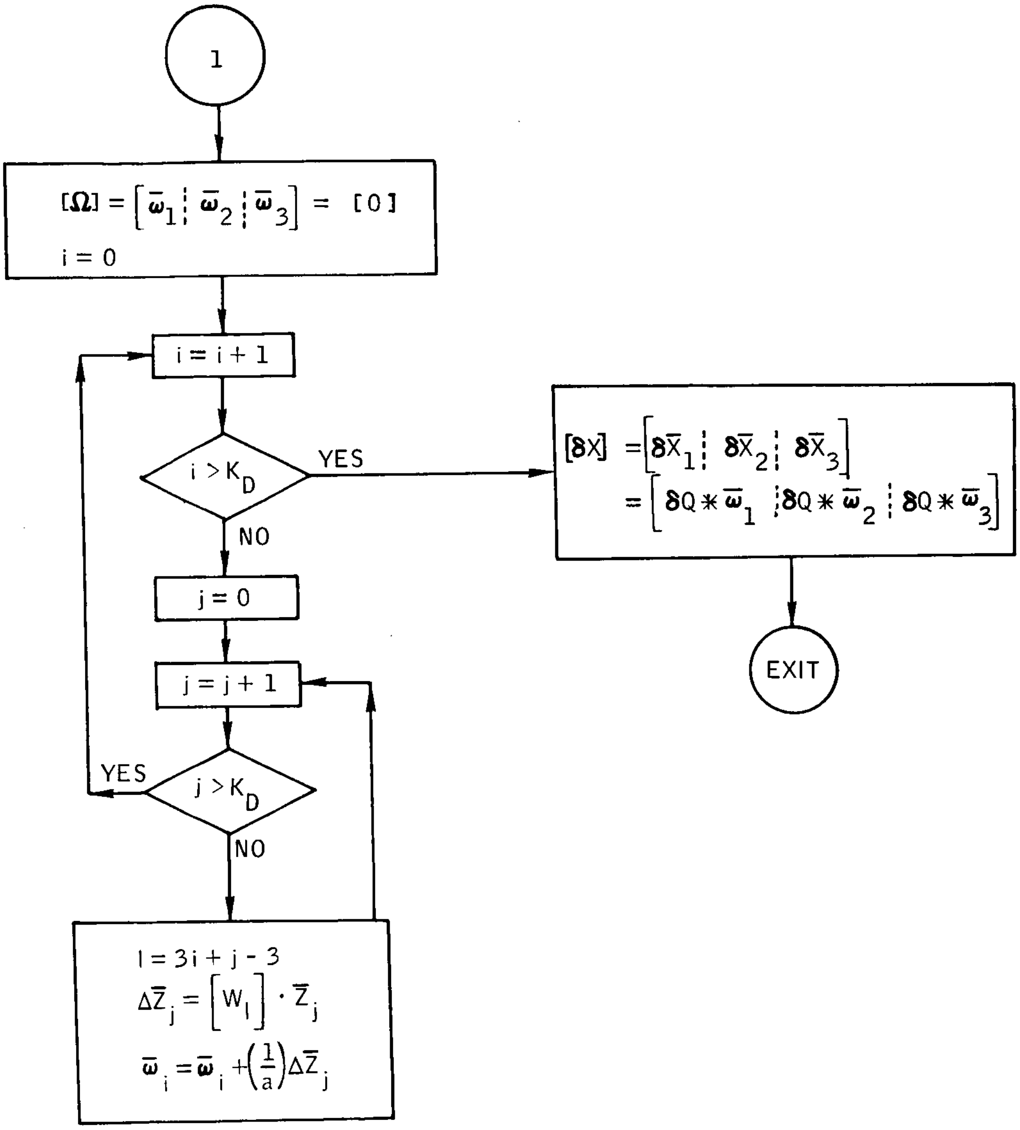
Page 14 of 14

Figure B-28.- Concluded.


ENTER
$K_D = \frac{J_D}{3}$
$[Z] = \left[\bar{Z}_1 \vdots \bar{Z}_2 \vdots \bar{Z}_3\right]$
$i = 0$
$i = i + 1$
$i > K_D$
YES
NO
$a = 0; \quad i = 0$
$j = 0$
$i = i + 1$
$j = j + 1$
$i:K_D$
$a = a + \bar{\alpha}^2$
YES
$j > K_D$
NO
$a = a + \bar{Z}_j \cdot \bar{Z}_j$
1
2
NO
$l = i + 3j - 3$
$\Delta\bar{Z}_j = \left[W_l\right]^T \cdot \bar{b}_j$
$\bar{Z}_i = \bar{Z}_i + \Delta\bar{Z}_j$

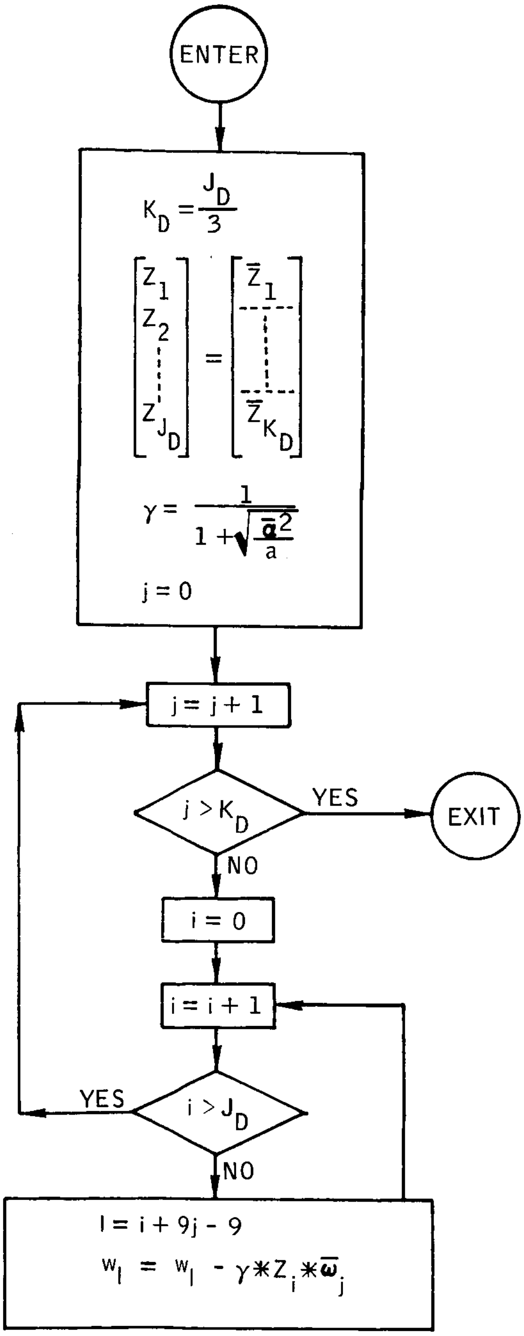
Page 1 of 2

Figure B-29.- Flow chart of subroutine DINCP1.


1
$[\Omega] = \left[\bar{\omega}_1 \vdots \bar{\omega}_2 \vdots \bar{\omega}_3\right] = [0]$
$i = 0$
$i = i + 1$
$i > K_D$
YES
NO
$[\delta X] = \left[\delta\bar{X}_1 \vdots \delta\bar{X}_2 \vdots \delta\bar{X}_3\right]$
$= \left[\delta Q * \bar{\omega}_1 \vdots \delta Q * \bar{\omega}_2 \vdots \delta Q * \bar{\omega}_3\right]$
EXIT
$j = 0$
$j = j + 1$
$j > K_D$
YES
NO
$l = 3i + j - 3$
$\Delta\bar{Z}_j = \left[W_l\right] \cdot \bar{Z}_j$
$\bar{\omega}_i = \bar{\omega}_i + \left(\frac{1}{a}\right)\Delta\bar{Z}_j$

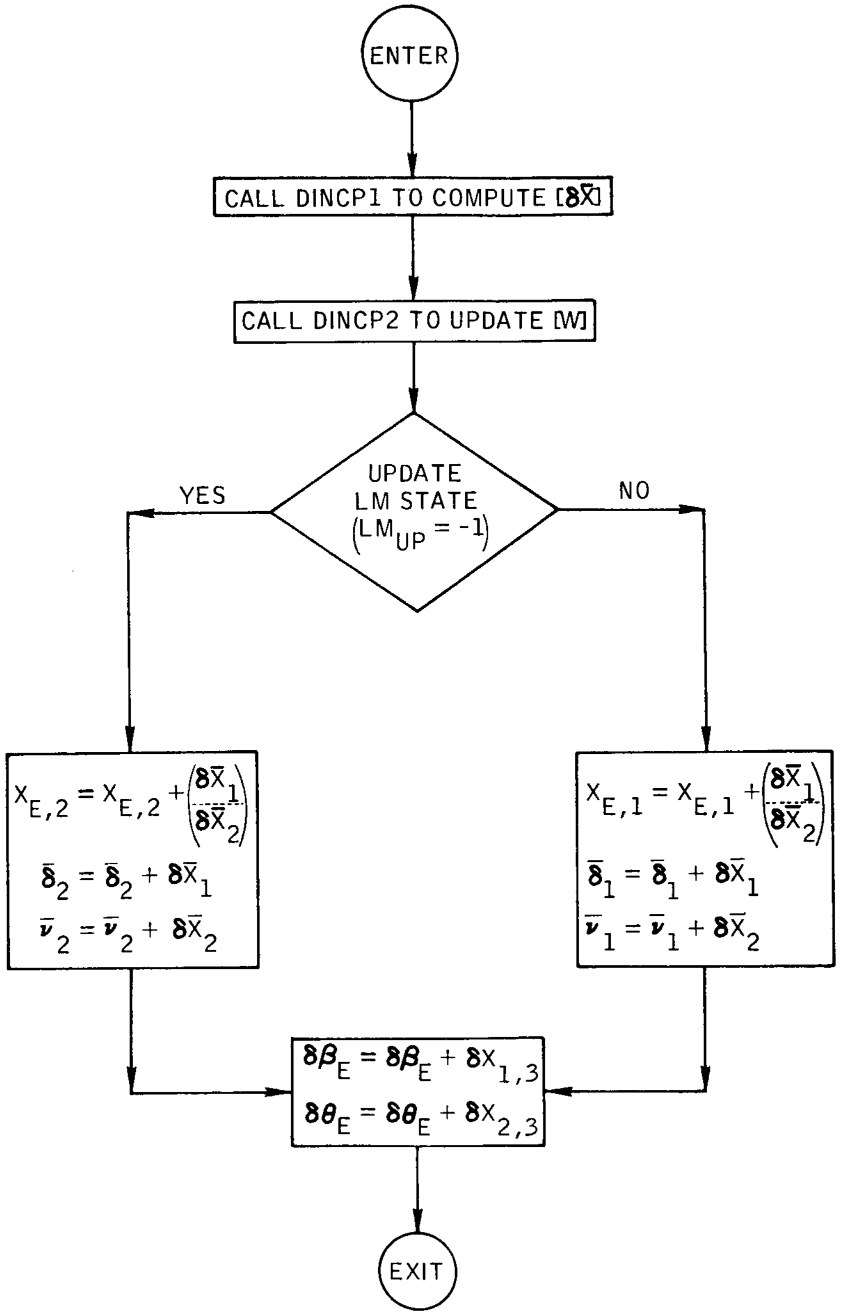
Page 2 of 2

Figure B-29.- Concluded.


ENTER
$K_D = \frac{J_D}{3}$
$\begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{J_D} \end{bmatrix} = \begin{bmatrix} \bar{Z}_1 \\ \vdots \\ \bar{Z}_{K_D} \end{bmatrix}$
$\gamma = \frac{1}{1 + \sqrt{\frac{\bar{a}^2}{a}}}$
$j = 0$
$j = j + 1$
$j > K_D$
YES
EXIT
NO
$i = 0$
$i = i + 1$
$i > J_D$
YES
NO
$l = i + 9j - 9$
$w_l = w_l - \gamma * Z_i * \bar{\omega}_j$

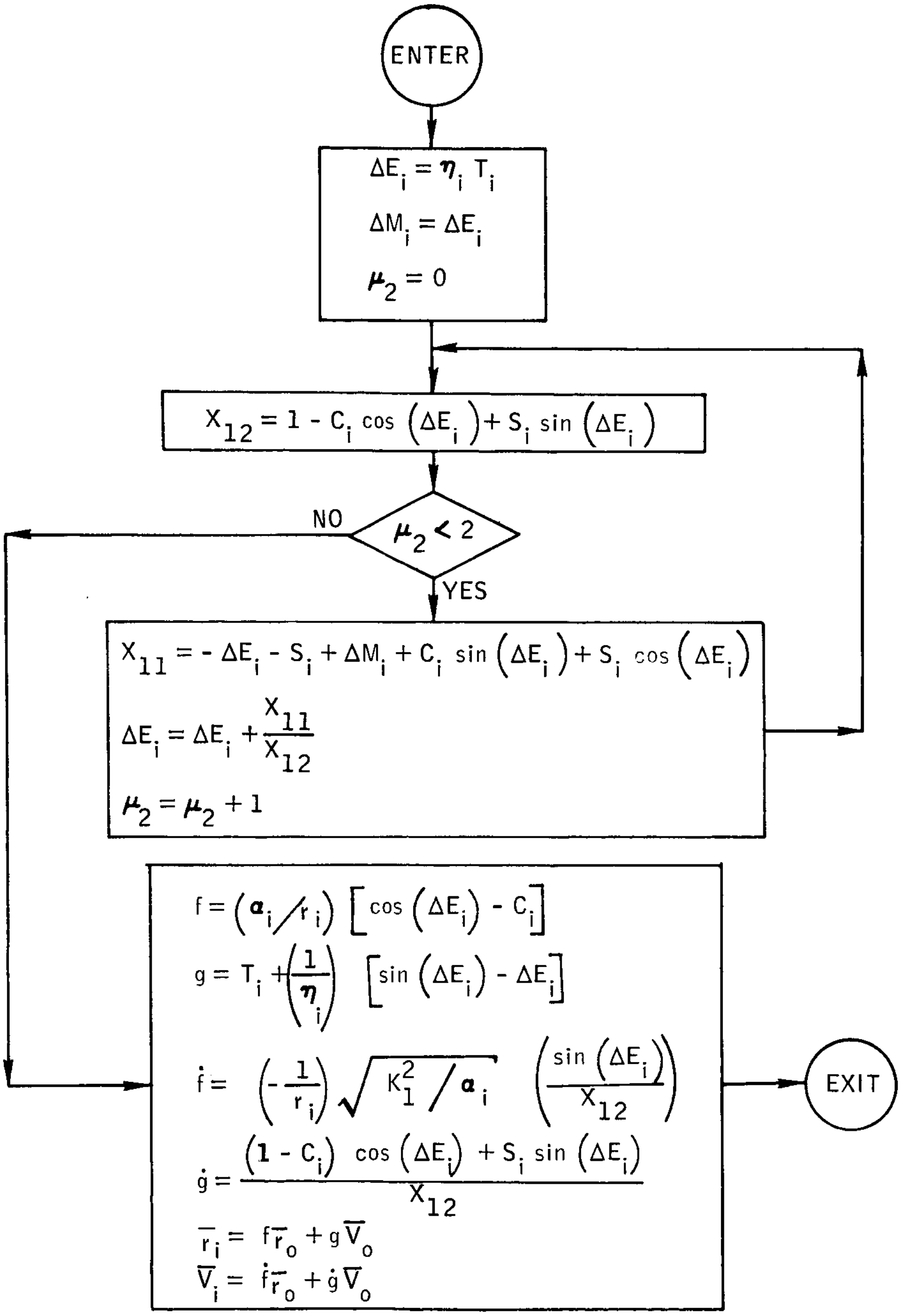
·Page 1 of 1

Figure B-30.- Flow chart of subroutine DINCP2.


ENTER
CALL DINCP1 TO COMPUTE $[\delta\bar{X}]$
CALL DINCP2 TO UPDATE [W]
UPDATE LM STATE $(LM_{UP} = -1)$
YES
NO
$X_{E,2} = X_{E,2} + \left(\frac{\delta\bar{X}_1}{\delta\bar{X}_2}\right)$
$\bar{\delta}_2 = \bar{\delta}_2 + \delta\bar{X}_1$
$\bar{\nu}_2 = \bar{\nu}_2 + \delta\bar{X}_2$
$X_{E,1} = X_{E,1} + \left(\frac{\delta\bar{X}_1}{\delta\bar{X}_2}\right)$
$\bar{\delta}_1 = \bar{\delta}_1 + \delta\bar{X}_1$
$\bar{\nu}_1 = \bar{\nu}_1 + \delta\bar{X}_2$
$\delta\beta_E = \delta\beta_E + \delta X_{1,3}$
$\delta\theta_E = \delta\theta_E + \delta X_{2,3}$
EXIT

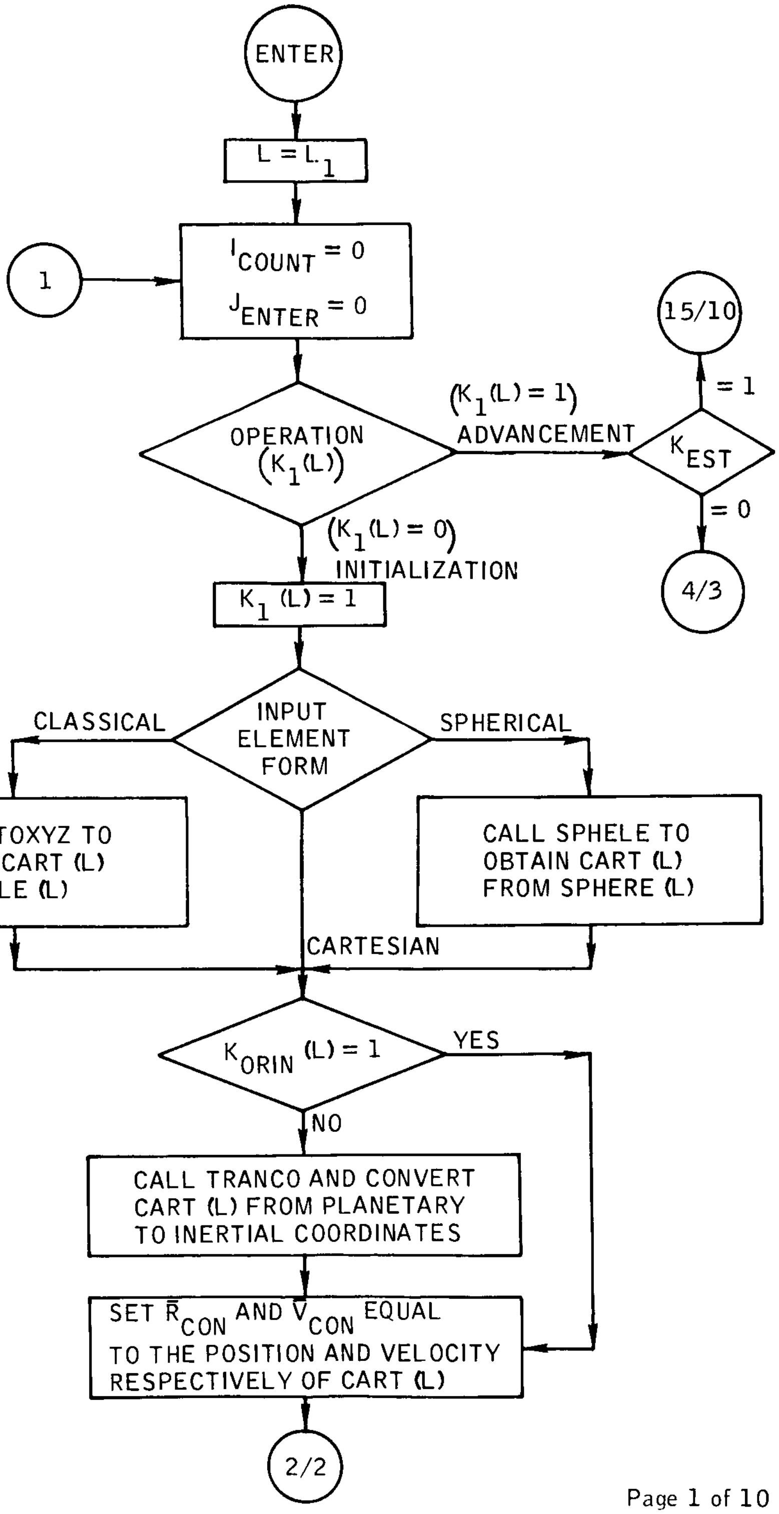
Page 1 of 1

Figure B-31.- Flow chart of subroutine DMESIP.


ENTER
$\Delta E_i = \eta_i T_i$
$\Delta M_i = \Delta E_i$
$\mu_2 = 0$
$X_{12} = 1 - C_i \cos\left(\Delta E_i\right) + S_i \sin\left(\Delta E_i\right)$
NO
$\mu_2 < 2$
YES
$X_{11} = -\Delta E_i - S_i + \Delta M_i + C_i \sin\left(\Delta E_i\right) + S_i \cos\left(\Delta E_i\right)$
$\Delta E_i = \Delta E_i + \frac{X_{11}}{X_{12}}$
$\mu_2 = \mu_2 + 1$
$f = \left(\alpha_i / r_i\right) \left[\cos\left(\Delta E_i\right) - C_i\right]$
$g = T_i + \left(\frac{1}{\eta_i}\right) \left[\sin\left(\Delta E_i\right) - \Delta E_i\right]$
$\dot{f} = \left(-\frac{1}{r_i}\right) \sqrt{K_1^2 / \alpha_i} \left(\frac{\sin\left(\Delta E_i\right)}{X_{12}}\right)$
$\dot{g} = \frac{\left(1 - C_i\right) \cos\left(\Delta E_i\right) + S_i \sin\left(\Delta E_i\right)}{X_{12}}$
$\bar{r}_i = f\bar{r}_o + g\bar{V}_o$
$\bar{V}_i = \dot{f}\bar{r}_o + \dot{g}\bar{V}_o$
EXIT

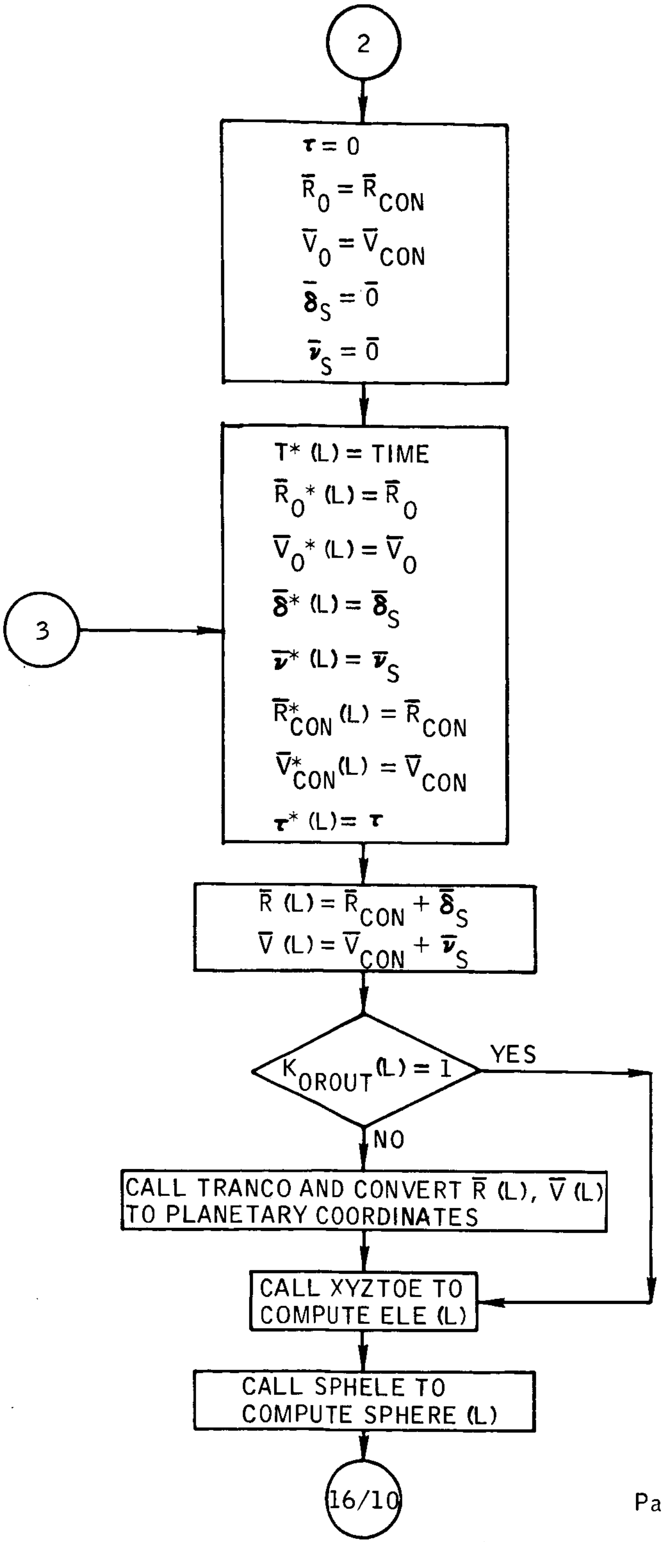
Page 1 of 1

Figure B-32.- Flow chart of subroutine ELPRE.


ENTER
L = L1
1
ICOUNT = 0
JENTER = 0
OPERATION (K1(L))
(K1(L) = 1) ADVANCEMENT
KEST
= 1
15/10
= 0
4/3
(K1(L) = 0) INITIALIZATION
K1 (L) = 1
INPUT ELEMENT FORM
CLASSICAL
SPHERICAL
CALL ETOXYZ TO OBTAIN CART (L) FROM ELE (L)
CALL SPHELE TO OBTAIN CART (L) FROM SPHERE (L)
CARTESIAN
KORIN (L) = 1
YES
NO
CALL TRANCO AND CONVERT CART (L) FROM PLANETARY TO INERTIAL COORDINATES
SET R̄CON AND V̄CON EQUAL TO THE POSITION AND VELOCITY RESPECTIVELY OF CART (L)
2/2

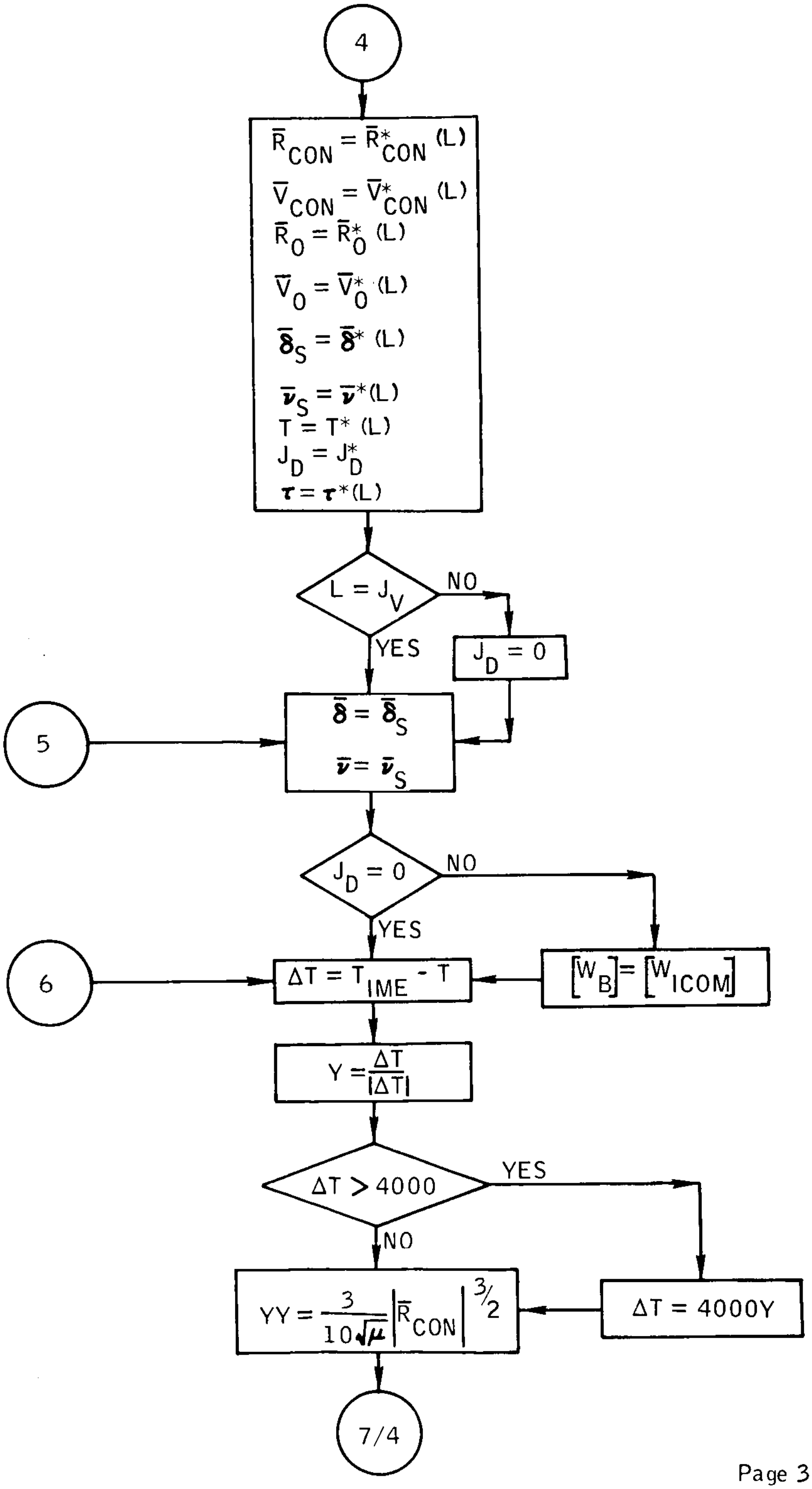
Page 1 of 10

Figure B-33.- Flow chart of subroutine ENCKE.


2
$\tau = 0$
$\bar{R}_0 = \bar{R}_{CON}$
$\bar{V}_0 = \bar{V}_{CON}$
$\bar{\delta}_S = \bar{0}$
$\bar{\nu}_S = \bar{0}$
3
$T^* (L) = TIME$
$\bar{R}_0{}^* (L) = \bar{R}_0$
$\bar{V}_0{}^* (L) = \bar{V}_0$
$\bar{\delta}^* (L) = \bar{\delta}_S$
$\bar{\nu}^* (L) = \bar{\nu}_S$
$\bar{R}^*_{CON} (L) = \bar{R}_{CON}$
$\bar{V}^*_{CON} (L) = \bar{V}_{CON}$
$\tau^* (L) = \tau$
$\bar{R} (L) = \bar{R}_{CON} + \bar{\delta}_S$
$\bar{V} (L) = \bar{V}_{CON} + \bar{\nu}_S$
$K_{OROUT} (L) = 1$
YES
NO
CALL TRANCO AND CONVERT $\bar{R}$ (L), $\bar{V}$ (L) TO PLANETARY COORDINATES
CALL XYZTOE TO COMPUTE ELE (L)
CALL SPHELE TO COMPUTE SPHERE (L)
16/10

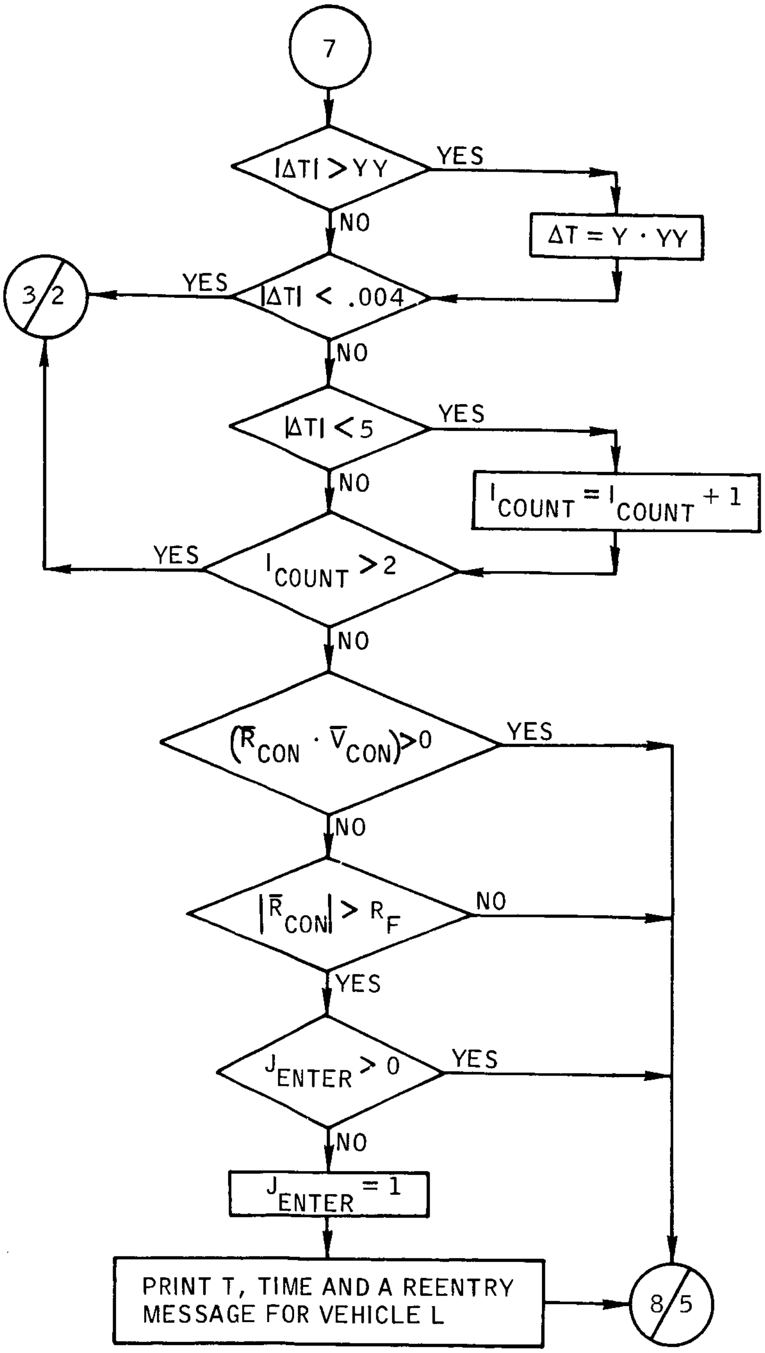
Page 2 of 10

Figure B-33.- Continued.


4
$\bar{R}_{CON} = \bar{R}^*_{CON}(L)$
$\bar{V}_{CON} = \bar{V}^*_{CON}(L)$
$\bar{R}_0 = \bar{R}^*_0(L)$
$\bar{V}_0 = \bar{V}^*_0(L)$
$\bar{\delta}_S = \bar{\delta}^*(L)$
$\bar{\nu}_S = \bar{\nu}^*(L)$
$T = T^*(L)$
$J_D = J^*_D$
$\tau = \tau^*(L)$
$L = J_V$
NO
YES
$J_D = 0$
5
$\bar{\delta} = \bar{\delta}_S$
$\bar{\nu} = \bar{\nu}_S$
$J_D = 0$
NO
YES
6
$\Delta T = T_{IME} - T$
$[W_B] = [W_{ICOM}]$
$Y = \frac{\Delta T}{|\Delta T|}$
$\Delta T > 4000$
YES
NO
$YY = \frac{3}{10\sqrt{\mu}} |\bar{R}_{CON}|^{3/2}$
$\Delta T = 4000Y$
7/4

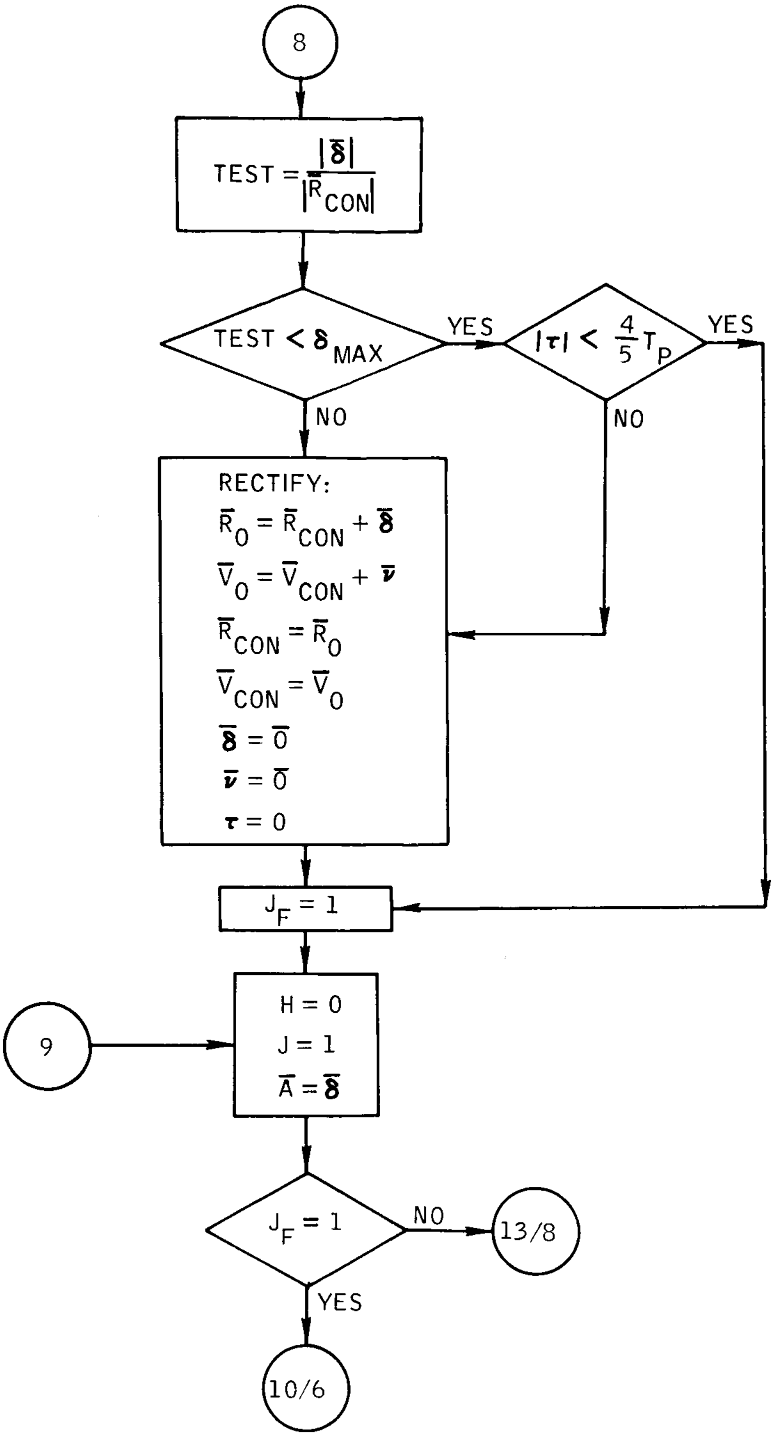
Page 3 of 10

Figure B-33.- Continued.



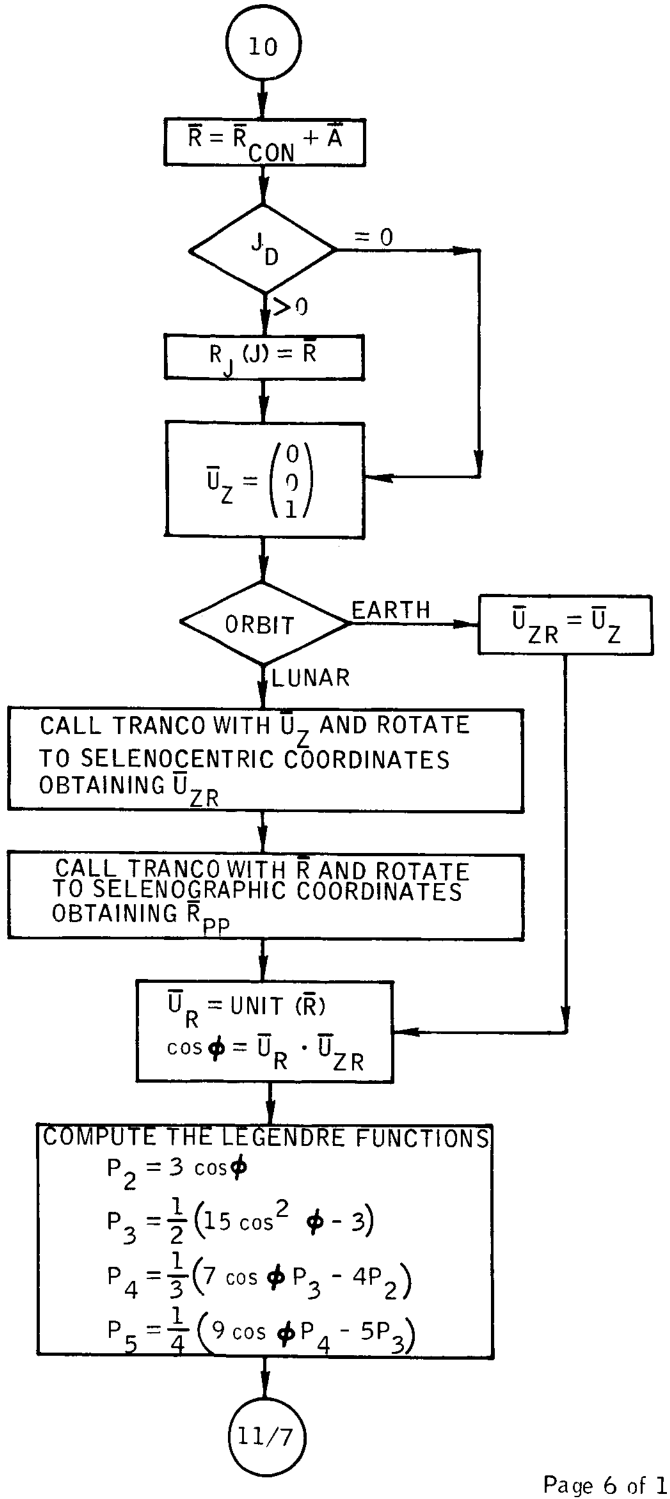
Page 4 of 10

Figure B-33.- Continued.


8
TEST = $\frac{|\bar{\delta}|}{|\bar{R}_{CON}|}$
TEST < δ_{MAX}
YES
$|\tau| < \frac{4}{5} T_P$
YES
NO
NO
RECTIFY:
$\bar{R}_0 = \bar{R}_{CON} + \bar{\delta}$
$\bar{V}_0 = \bar{V}_{CON} + \bar{\nu}$
$\bar{R}_{CON} = \bar{R}_0$
$\bar{V}_{CON} = \bar{V}_0$
$\bar{\delta} = \bar{0}$
$\bar{\nu} = \bar{0}$
$\tau = 0$
$J_F = 1$
9
H = 0
J = 1
$\bar{A} = \bar{\delta}$
$J_F = 1$
NO
13/8
YES
10/6

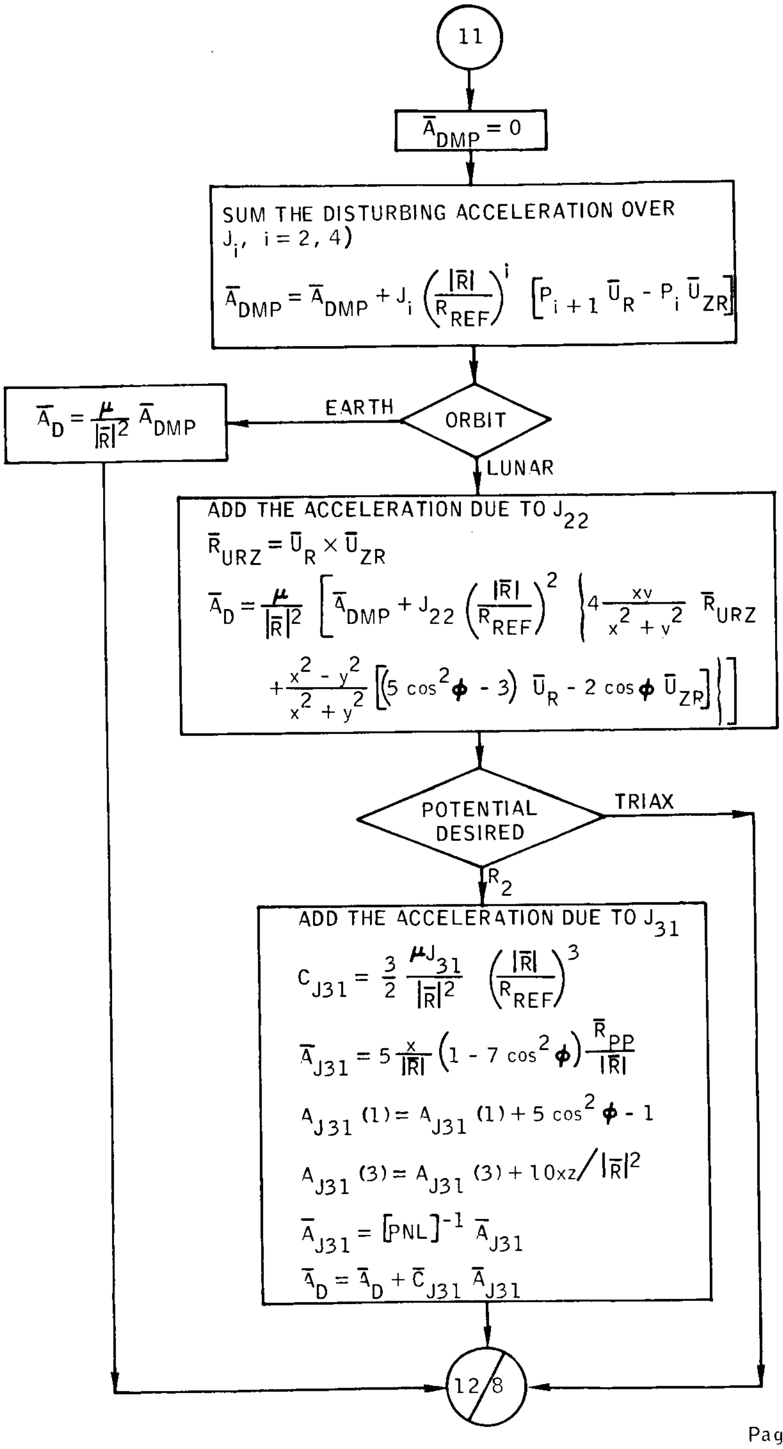
Page 5 of 10

Figure B-33.- Continued.


10
$\bar{R} = \bar{R}_{CON} + \bar{A}$
J_D
= 0
> 0
$R_J(J) = \bar{R}$
$\bar{U}_Z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$
ORBIT
EARTH
$\bar{U}_{ZR} = \bar{U}_Z$
LUNAR
CALL TRANCO WITH $\bar{U}_Z$ AND ROTATE TO SELENOCENTRIC COORDINATES OBTAINING $\bar{U}_{ZR}$
CALL TRANCO WITH $\bar{R}$ AND ROTATE TO SELENOGRAPHIC COORDINATES OBTAINING $\bar{R}_{PP}$
$\bar{U}_R = \text{UNIT}(\bar{R})$
$\cos\phi = \bar{U}_R \cdot \bar{U}_{ZR}$
COMPUTE THE LEGENDRE FUNCTIONS
$P_2 = 3 \cos\phi$
$P_3 = \frac{1}{2}(15 \cos^2 \phi - 3)$
$P_4 = \frac{1}{3}(7 \cos\phi P_3 - 4P_2)$
$P_5 = \frac{1}{4}(9 \cos\phi P_4 - 5P_3)$
11/7

Page 6 of 10

Figure B-33.- Continued.


11
$\bar{A}_{DMP} = 0$
SUM THE DISTURBING ACCELERATION OVER J_i, $i = 2, 4)$
$\bar{A}_{DMP} = \bar{A}_{DMP} + J_i \left(\frac{|\bar{R}|}{R_{REF}}\right)^i \left[P_{i+1} \bar{U}_R - P_i \bar{U}_{ZR}\right]$
EARTH
ORBIT
LUNAR
$\bar{A}_D = \frac{\mu}{|\bar{R}|^2} \bar{A}_{DMP}$
ADD THE ACCELERATION DUE TO J_{22}
$\bar{R}_{URZ} = \bar{U}_R \times \bar{U}_{ZR}$
$\bar{A}_D = \frac{\mu}{|\bar{R}|^2} \left[\bar{A}_{DMP} + J_{22} \left(\frac{|\bar{R}|}{R_{REF}}\right)^2 \left\{4 \frac{xv}{x^2 + v^2} \bar{R}_{URZ} + \frac{x^2 - y^2}{x^2 + y^2} \left[(5 \cos^2 \phi - 3) \bar{U}_R - 2 \cos \phi \, \bar{U}_{ZP}\right]\right\}\right]$
POTENTIAL DESIRED
TRIAX
R_2
ADD THE ACCELERATION DUE TO J_{31}
$C_{J31} = \frac{3}{2} \frac{\mu J_{31}}{|\bar{R}|^2} \left(\frac{|\bar{R}|}{R_{REF}}\right)^3$
$\bar{A}_{J31} = 5 \frac{x}{|\bar{R}|} \left(1 - 7 \cos^2 \phi\right) \frac{\bar{R}_{PP}}{|\bar{R}|}$
$A_{J31}(1) = A_{J31}(1) + 5 \cos^2 \phi - 1$
$A_{J31}(3) = A_{J31}(3) + 10xz/|\bar{R}|^2$
$\bar{A}_{J31} = [PNL]^{-1} \bar{A}_{J31}$
$\bar{A}_D = \bar{A}_D + \bar{C}_{J31} \bar{A}_{J31}$
12/8

Page 7 of 10

Figure B-33.- Continued.

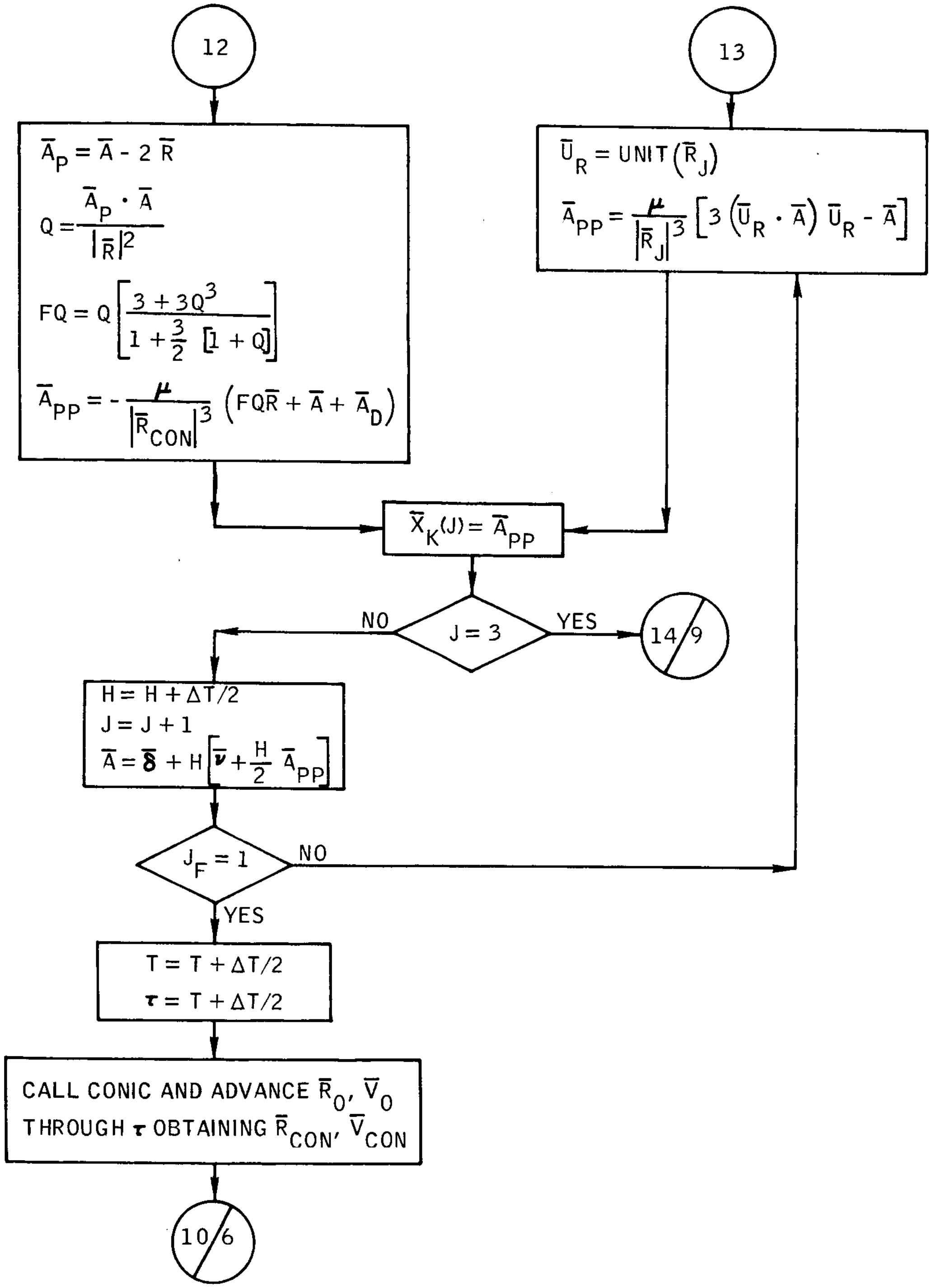
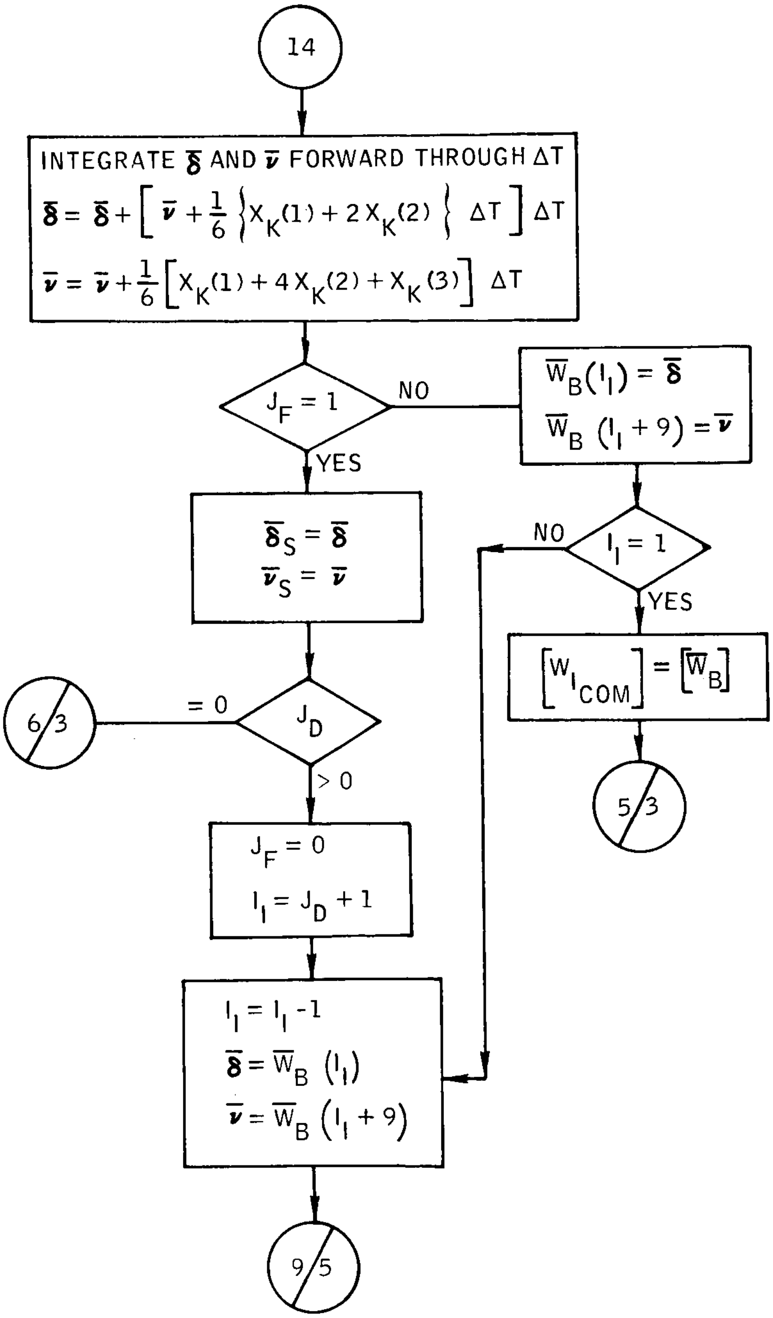

12
13
$\bar{A}_P = \bar{A} - 2\,\bar{R}$
$Q = \frac{\bar{A}_P \cdot \bar{A}}{|\bar{R}|^2}$
$FQ = Q\left[\frac{3 + 3Q^3}{1 + \frac{3}{2}\,[1 + Q]}\right]$
$\bar{A}_{PP} = -\frac{\mu}{|\bar{R}_{CON}|^3}\left(FQ\bar{R} + \bar{A} + \bar{A}_D\right)$
$\bar{U}_R = \text{UNIT}\left(\bar{R}_J\right)$
$\bar{A}_{PP} = \frac{\mu}{|\bar{R}_J|^3}\left[3\left(\bar{U}_R \cdot \bar{A}\right)\bar{U}_R - \bar{A}\right]$
$\bar{X}_K(J) = \bar{A}_{PP}$
NO
$J = 3$
YES
14/9
$H = H + \Delta T/2$
$J = J + 1$
$\bar{A} = \bar{\delta} + H\left[\bar{\nu} + \frac{H}{2}\,\bar{A}_{PP}\right]$
$J_F = 1$
NO
YES
$T = T + \Delta T/2$
$\tau = T + \Delta T/2$
CALL CONIC AND ADVANCE $\bar{R}_0, \bar{V}_0$
THROUGH τ OBTAINING $\bar{R}_{CON}, \bar{V}_{CON}$
10/6

Figure B-33.- Continued.


14
INTEGRATE $\bar{\delta}$ AND $\bar{\nu}$ FORWARD THROUGH ΔT
$\bar{\delta} = \bar{\delta} + \left[\bar{\nu} + \frac{1}{6} \left\{ X_K(1) + 2X_K(2) \right\} \Delta T \right] \Delta T$
$\bar{\nu} = \bar{\nu} + \frac{1}{6} \left[X_K(1) + 4X_K(2) + X_K(3) \right] \Delta T$
$J_F = 1$
NO
YES
$\bar{W}_B(I_I) = \bar{\delta}$
$\bar{W}_B(I_I + 9) = \bar{\nu}$
$\bar{\delta}_S = \bar{\delta}$
$\bar{\nu}_S = \bar{\nu}$
$I_I = 1$
NO
YES
$\left[W_{I_{COM}}\right] = \left[\bar{W}_B\right]$
5/3
6/3
= 0
J_D
> 0
$J_F = 0$
$I_I = J_D + 1$
$I_I = I_I - 1$
$\bar{\delta} = \bar{W}_B(I_I)$
$\bar{\nu} = \bar{W}_B(I_I + 9)$
9/5

Page 9 of 10

Figure B-33.- Continued.

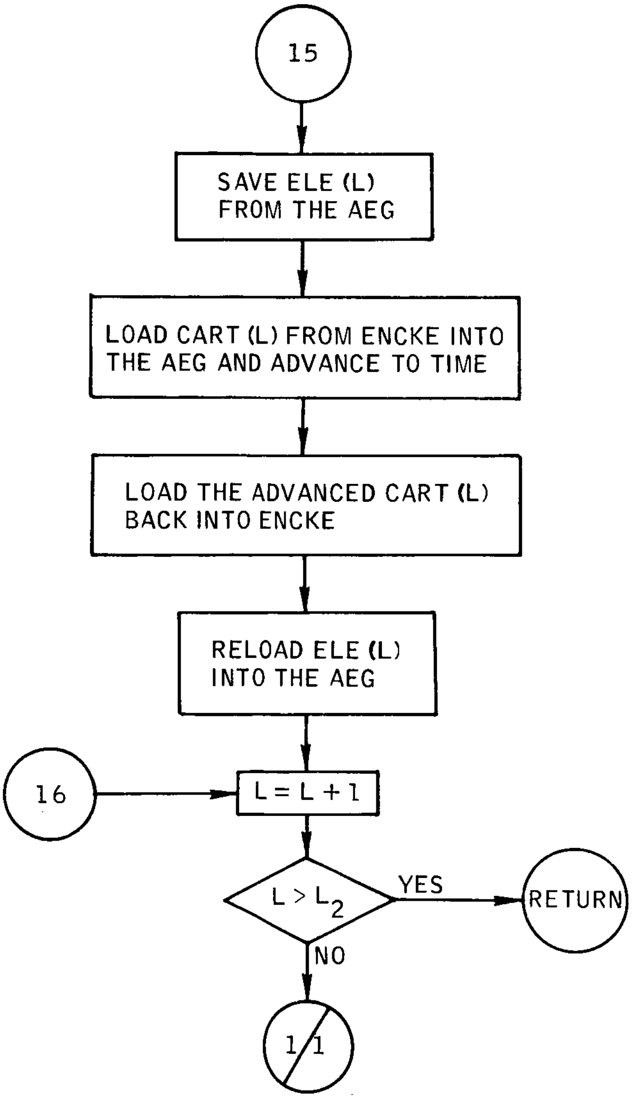
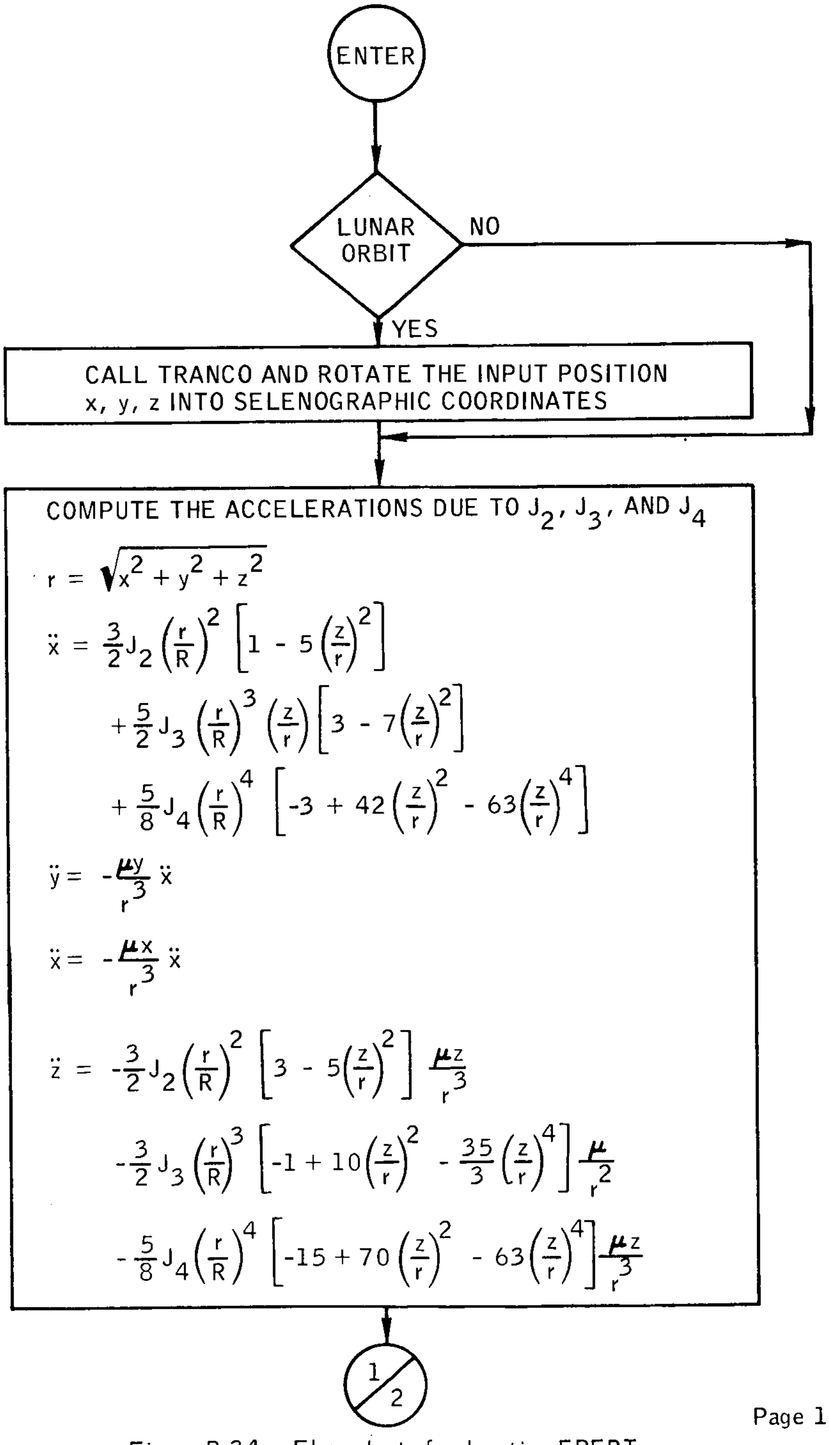

15
SAVE ELE (L) FROM THE AEG
LOAD CART (L) FROM ENCKE INTO THE AEG AND ADVANCE TO TIME
LOAD THE ADVANCED CART (L) BACK INTO ENCKE
RELOAD ELE (L) INTO THE AEG
16
L = L + 1
L > L2
YES
RETURN
NO
1/1

Figure B-33.- Concluded. Page 10 of 10



CALL TRANCO AND ROTATE THE INPUT POSITION x, y, z INTO SELENOGRAPHIC COORDINATES

COMPUTE THE ACCELERATIONS DUE TO J_2, J_3, AND J_4

$$r = \sqrt{x^2 + y^2 + z^2}$$

$$\ddot{x} = \frac{3}{2}J_2\left(\frac{r}{R}\right)^2\left[1 - 5\left(\frac{z}{r}\right)^2\right] + \frac{5}{2}J_3\left(\frac{r}{R}\right)^3\left(\frac{z}{r}\right)\left[3 - 7\left(\frac{z}{r}\right)^2\right] + \frac{5}{8}J_4\left(\frac{r}{R}\right)^4\left[-3 + 42\left(\frac{z}{r}\right)^2 - 63\left(\frac{z}{r}\right)^4\right]$$

$$\ddot{y} = -\frac{\mu y}{r^3}\ddot{x}$$

$$\ddot{x} = -\frac{\mu x}{r^3}\ddot{x}$$

$$\ddot{z} = -\frac{3}{2}J_2\left(\frac{r}{R}\right)^2\left[3 - 5\left(\frac{z}{r}\right)^2\right]\frac{\mu z}{r^3} - \frac{3}{2}J_3\left(\frac{r}{R}\right)^3\left[-1 + 10\left(\frac{z}{r}\right)^2 - \frac{35}{3}\left(\frac{z}{r}\right)^4\right]\frac{\mu}{r^2} - \frac{5}{8}J_4\left(\frac{r}{R}\right)^4\left[-15 + 70\left(\frac{z}{r}\right)^2 - 63\left(\frac{z}{r}\right)^4\right]\frac{\mu z}{r^3}$$

Page 1 of 2

Figure B-34.- Flow chart of subroutine EPERT.

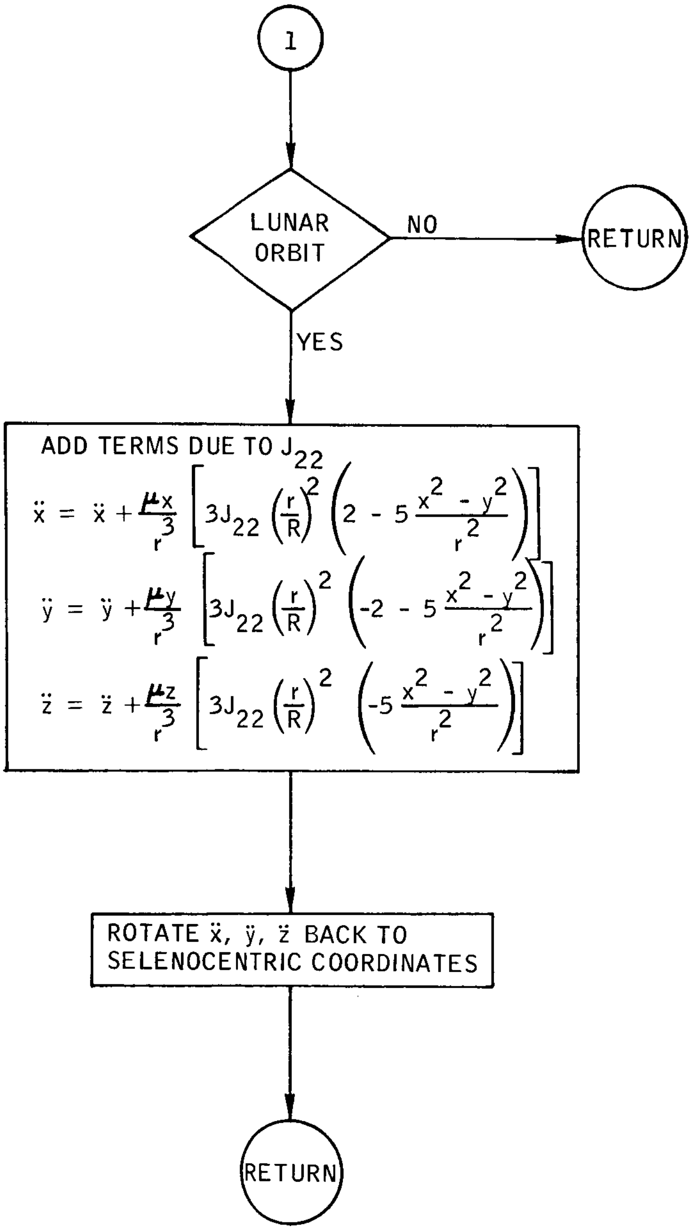

1
LUNAR ORBIT
NO
RETURN
YES
ADD TERMS DUE TO J_{22}
$\ddot{x} = \ddot{x} + \frac{\mu x}{r^3}\left[3J_{22}\left(\frac{r}{R}\right)^2\left(2 - 5\frac{x^2 - y^2}{r^2}\right)\right]$
$\ddot{y} = \ddot{y} + \frac{\mu y}{r^3}\left[3J_{22}\left(\frac{r}{R}\right)^2\left(-2 - 5\frac{x^2 - y^2}{r^2}\right)\right]$
$\ddot{z} = \ddot{z} + \frac{\mu z}{r^3}\left[3J_{22}\left(\frac{r}{R}\right)^2\left(-5\frac{x^2 - y^2}{r^2}\right)\right]$
ROTATE $\ddot{x}$, $\ddot{y}$, $\ddot{z}$ BACK TO SELENOCENTRIC COORDINATES
RETURN

Figure B-34.- Concluded.

Page 2 of 2

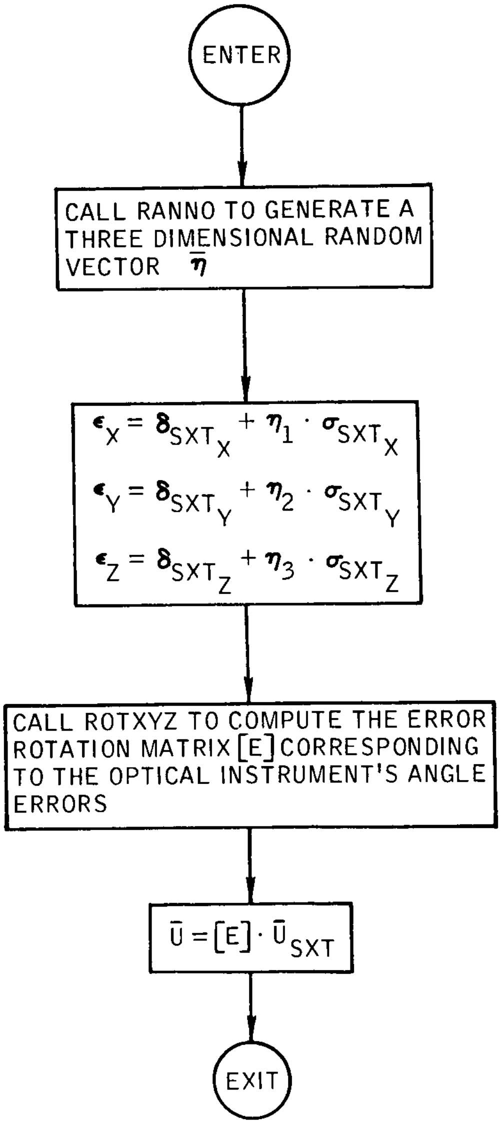
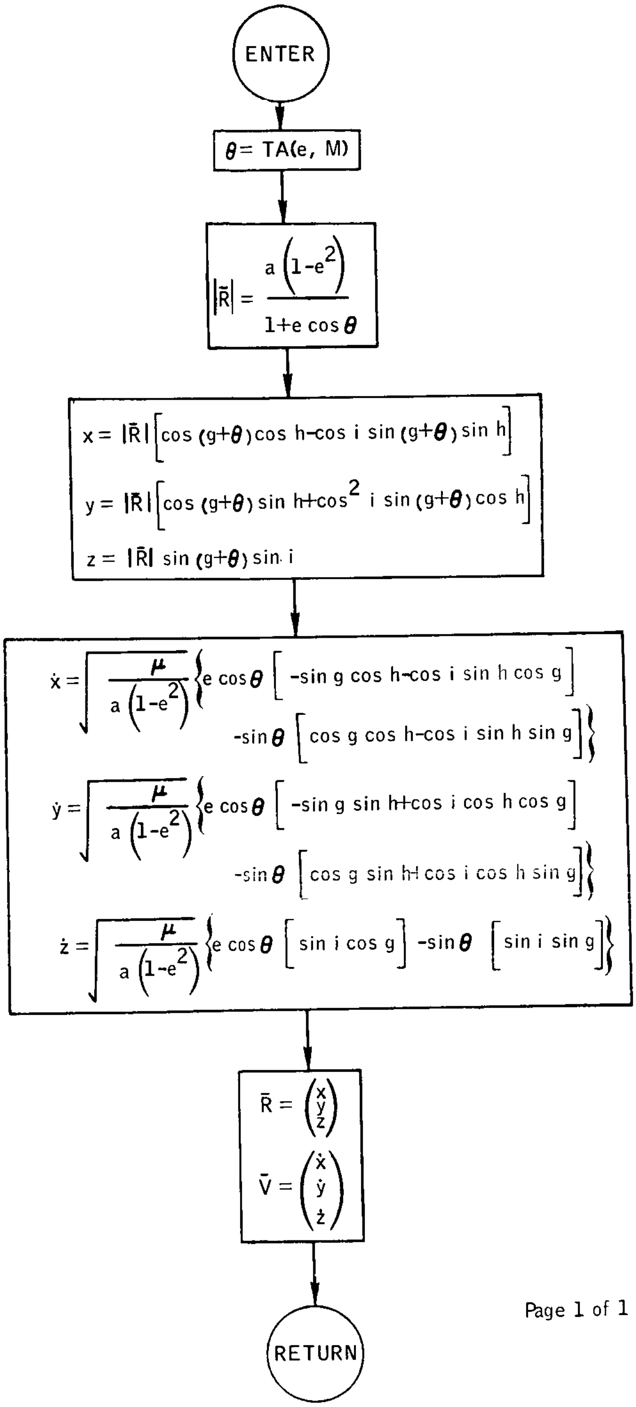

ENTER
CALL RANNO TO GENERATE A THREE DIMENSIONAL RANDOM VECTOR $\bar{\eta}$
$\epsilon_X = \delta_{SXT_X} + \eta_1 \cdot \sigma_{SXT_X}$
$\epsilon_Y = \delta_{SXT_Y} + \eta_2 \cdot \sigma_{SXT_Y}$
$\epsilon_Z = \delta_{SXT_Z} + \eta_3 \cdot \sigma_{SXT_Z}$
CALL ROTXYZ TO COMPUTE THE ERROR ROTATION MATRIX [E] CORRESPONDING TO THE OPTICAL INSTRUMENT'S ANGLE ERRORS
$\bar{U} = [E] \cdot \bar{U}_{SXT}$
EXIT

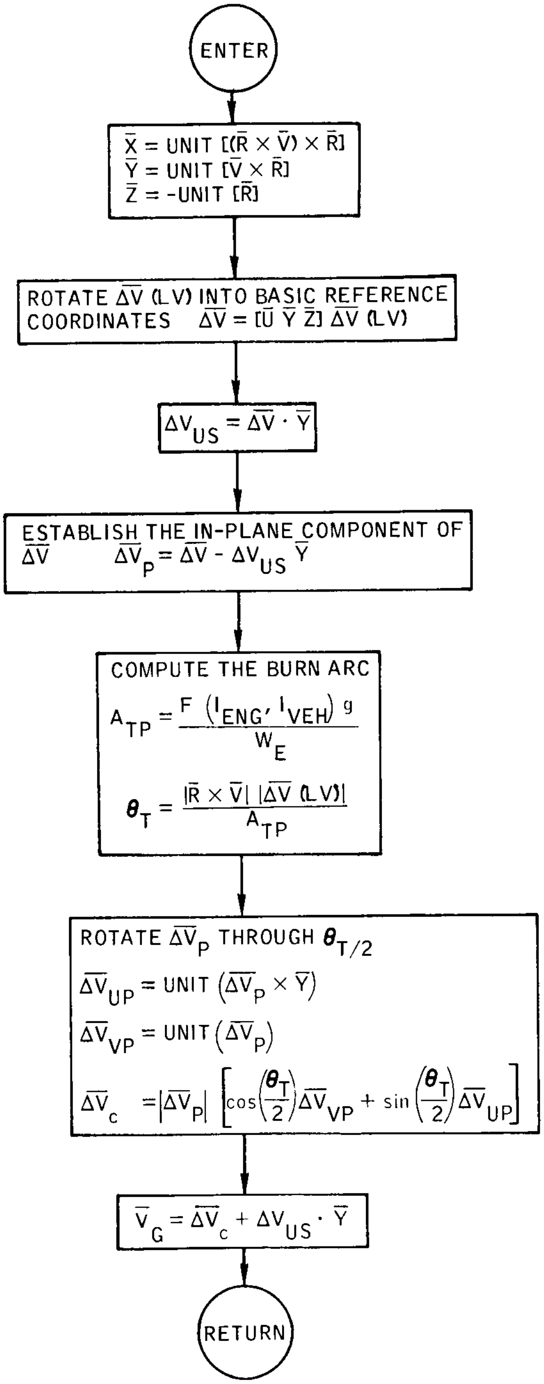
Figure B-35.- Flow chart of subroutine ERRSXT.

Page 1 of 1


ENTER
$\theta = TA(e, M)$
$|\bar{R}| = \dfrac{a\left(1-e^2\right)}{1+e\cos\theta}$
$x = |\bar{R}|\left[\cos(g+\theta)\cos h - \cos i \sin(g+\theta)\sin h\right]$
$y = |\bar{R}|\left[\cos(g+\theta)\sin h + \cos^2 i \sin(g+\theta)\cos h\right]$
$z = |\bar{R}| \sin(g+\theta)\sin i$
$\dot{x} = \sqrt{\dfrac{\mu}{a\left(1-e^2\right)}}\left\{e\cos\theta\left[-\sin g\cos h - \cos i \sin h \cos g\right] - \sin\theta\left[\cos g \cos h - \cos i \sin h \sin g\right]\right\}$
$\dot{y} = \sqrt{\dfrac{\mu}{a\left(1-e^2\right)}}\left\{e\cos\theta\left[-\sin g\sin h + \cos i \cos h \cos g\right] - \sin\theta\left[\cos g \sin h + \cos i \cos h \sin g\right]\right\}$
$\dot{z} = \sqrt{\dfrac{\mu}{a\left(1-e^2\right)}}\left\{e\cos\theta\left[\sin i \cos g\right] - \sin\theta\left[\sin i \sin g\right]\right\}$
$\bar{R} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$
$\bar{V} = \begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{pmatrix}$
RETURN

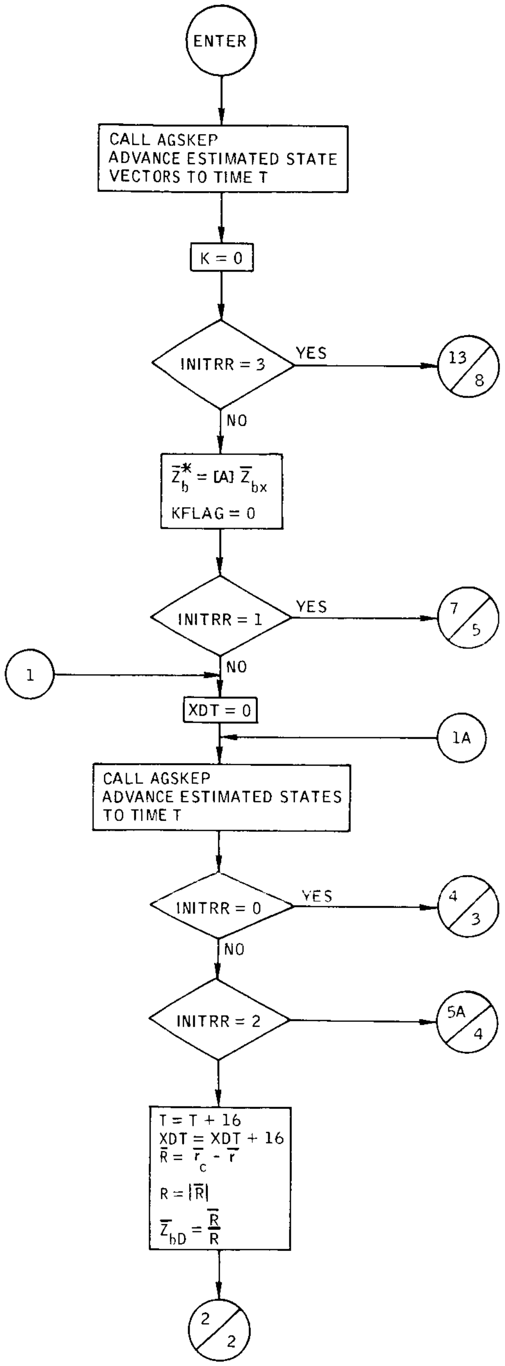
Page 1 of 1

Figure B-36.- Flow chart of subroutine ETOXYZ.


ENTER
$\bar{X}$ = UNIT $[(\bar{R} \times \bar{V}) \times \bar{R}]$
$\bar{Y}$ = UNIT $[\bar{V} \times \bar{R}]$
$\bar{Z}$ = -UNIT $[\bar{R}]$
ROTATE $\overline{\Delta V}$ (LV) INTO BASIC REFERENCE COORDINATES $\overline{\Delta V} = [\bar{U}\ \bar{Y}\ \bar{Z}]\ \overline{\Delta V}$ (LV)
$\Delta V_{US} = \overline{\Delta V} \cdot \bar{Y}$
ESTABLISH THE IN-PLANE COMPONENT OF $\overline{\Delta V}$ $\overline{\Delta V}_P = \overline{\Delta V} - \Delta V_{US}\ \bar{Y}$
COMPUTE THE BURN ARC
$A_{TP} = \frac{F\ (I_{ENG},\ I_{VEH})\ g}{W_E}$
$\theta_T = \frac{|\bar{R} \times \bar{V}|\ |\overline{\Delta V}\ (LV)|}{A_{TP}}$
ROTATE $\overline{\Delta V}_P$ THROUGH $\theta_{T/2}$
$\overline{\Delta V}_{UP}$ = UNIT $\left(\overline{\Delta V}_P \times \bar{Y}\right)$
$\overline{\Delta V}_{VP}$ = UNIT $\left(\overline{\Delta V}_P\right)$
$\overline{\Delta V}_c = |\overline{\Delta V}_P| \left[\cos\left(\frac{\theta_T}{2}\right)\overline{\Delta V}_{VP} + \sin\left(\frac{\theta_T}{2}\right)\overline{\Delta V}_{UP}\right]$
$\bar{V}_G = \overline{\Delta V}_c + \Delta V_{US} \cdot \bar{Y}$
RETURN

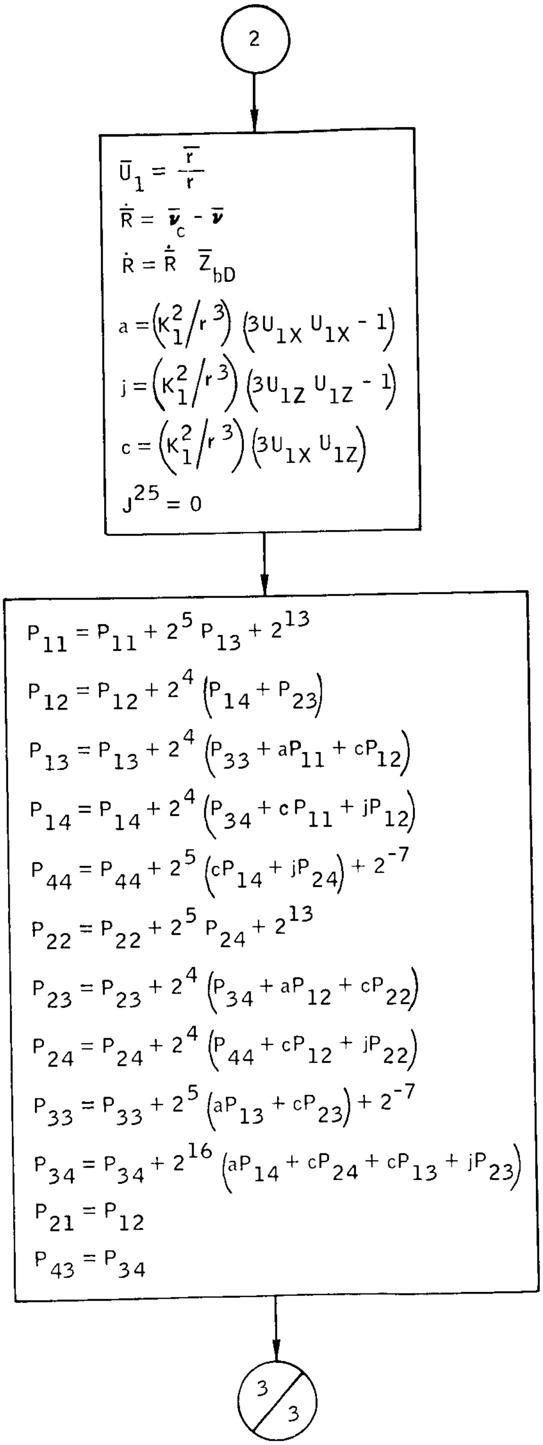
Page 1 of 1

Figure B-37.- Flow chart of subroutine EXDV.



Page 1 of 8

Figure B-38.- Flow chart of subroutine FILTER.



$\bar{U}_1 = \frac{\bar{r}}{r}$

$\dot{\bar{R}} = \bar{\boldsymbol{\nu}}_c - \bar{\boldsymbol{\nu}}$

$\dot{R} = \dot{\bar{R}} \quad \bar{Z}_{bD}$

$a = \left(K_1^2 / r^3\right) \left(3U_{1X} U_{1X} - 1\right)$

$j = \left(K_1^2 / r^3\right) \left(3U_{1Z} U_{1Z} - 1\right)$

$c = \left(K_1^2 / r^3\right) \left(3U_{1X} U_{1Z}\right)$

$J^{25} = 0$

$P_{11} = P_{11} + 2^5 P_{13} + 2^{13}$

$P_{12} = P_{12} + 2^4 \left(P_{14} + P_{23}\right)$

$P_{13} = P_{13} + 2^4 \left(P_{33} + aP_{11} + cP_{12}\right)$

$P_{14} = P_{14} + 2^4 \left(P_{34} + cP_{11} + jP_{12}\right)$

$P_{44} = P_{44} + 2^5 \left(cP_{14} + jP_{24}\right) + 2^{-7}$

$P_{22} = P_{22} + 2^5 P_{24} + 2^{13}$

$P_{23} = P_{23} + 2^4 \left(P_{34} + aP_{12} + cP_{22}\right)$

$P_{24} = P_{24} + 2^4 \left(P_{44} + cP_{12} + jP_{22}\right)$

$P_{33} = P_{33} + 2^5 \left(aP_{13} + cP_{23}\right) + 2^{-7}$

$P_{34} = P_{34} + 2^{16} \left(aP_{14} + cP_{24} + cP_{13} + jP_{23}\right)$

$P_{21} = P_{12}$

$P_{43} = P_{34}$

Figure B-38.- Continued.

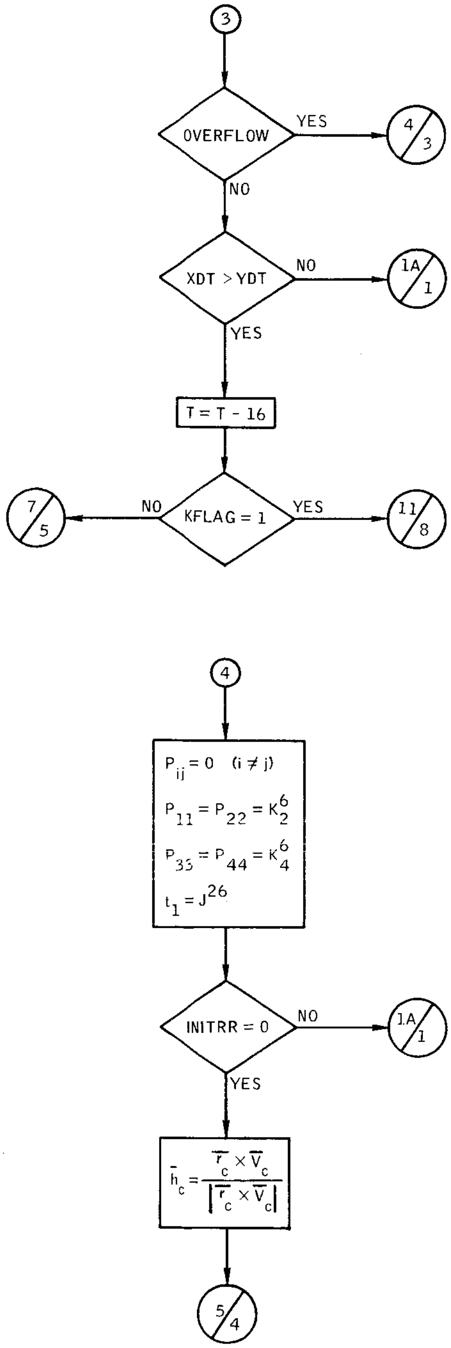
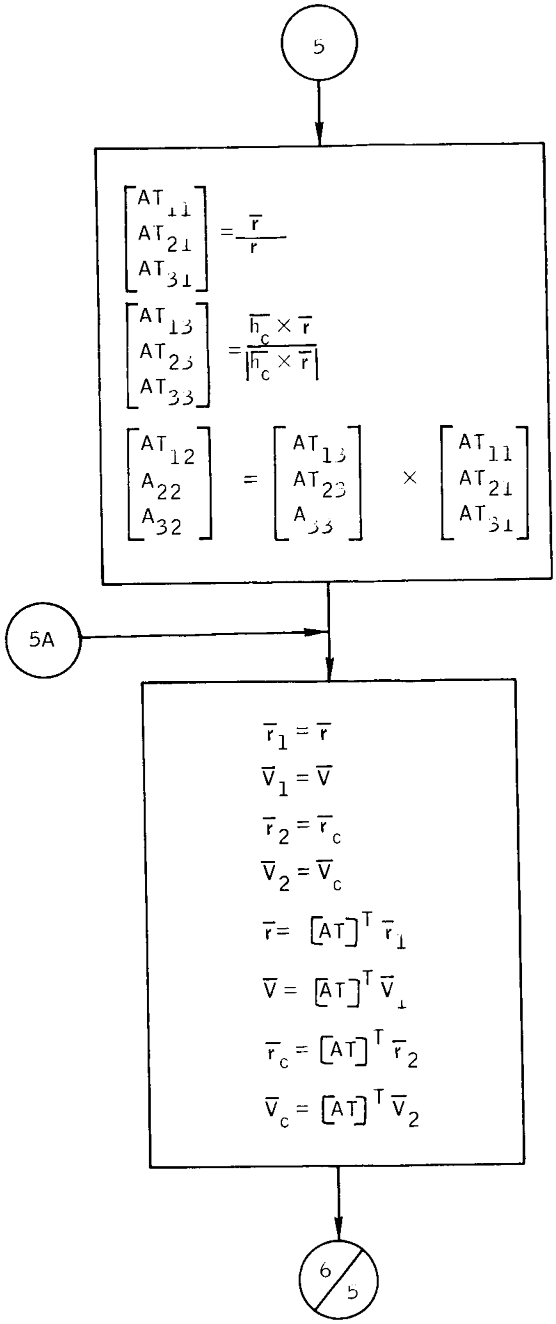
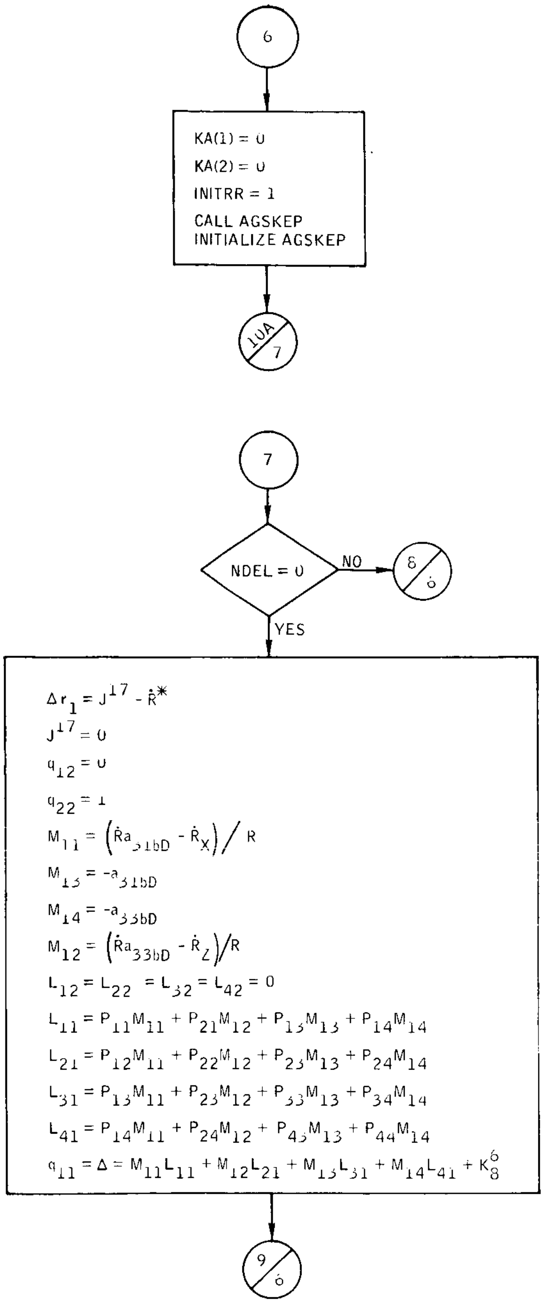

3
OVERFLOW
YES
4/3
NO
XDT > YDT
NO
1A/1
YES
T = T - 16
KFLAG = 1
NO
7/5
YES
11/8
4
$P_{ij} = 0 \quad (i \neq j)$
$P_{11} = P_{22} = K_2^6$
$P_{33} = P_{44} = K_4^6$
$t_1 = J^{26}$
INITRR = 0
NO
1A/1
YES
$\bar{h}_c = \frac{\bar{r}_c \times \bar{V}_c}{|\bar{r}_c \times \bar{V}_c|}$
5/4

Figure B-38.- Continued.


5
$\begin{bmatrix} AT_{11} \\ AT_{21} \\ AT_{31} \end{bmatrix} = \frac{\bar{r}}{r}$
$\begin{bmatrix} AT_{13} \\ AT_{23} \\ AT_{33} \end{bmatrix} = \frac{\bar{h}_c \times \bar{r}}{|\bar{h}_c \times \bar{r}|}$
$\begin{bmatrix} AT_{12} \\ A_{22} \\ A_{32} \end{bmatrix} = \begin{bmatrix} AT_{13} \\ AT_{23} \\ A_{33} \end{bmatrix} \times \begin{bmatrix} AT_{11} \\ AT_{21} \\ AT_{31} \end{bmatrix}$
5A
$\bar{r}_1 = \bar{r}$
$\bar{V}_1 = \bar{V}$
$\bar{r}_2 = \bar{r}_c$
$\bar{V}_2 = \bar{V}_c$
$\bar{r} = [AT]^T \bar{r}_1$
$\bar{V} = [AT]^T \bar{V}_1$
$\bar{r}_c = [AT]^T \bar{r}_2$
$\bar{V}_c = [AT]^T \bar{V}_2$
6/5

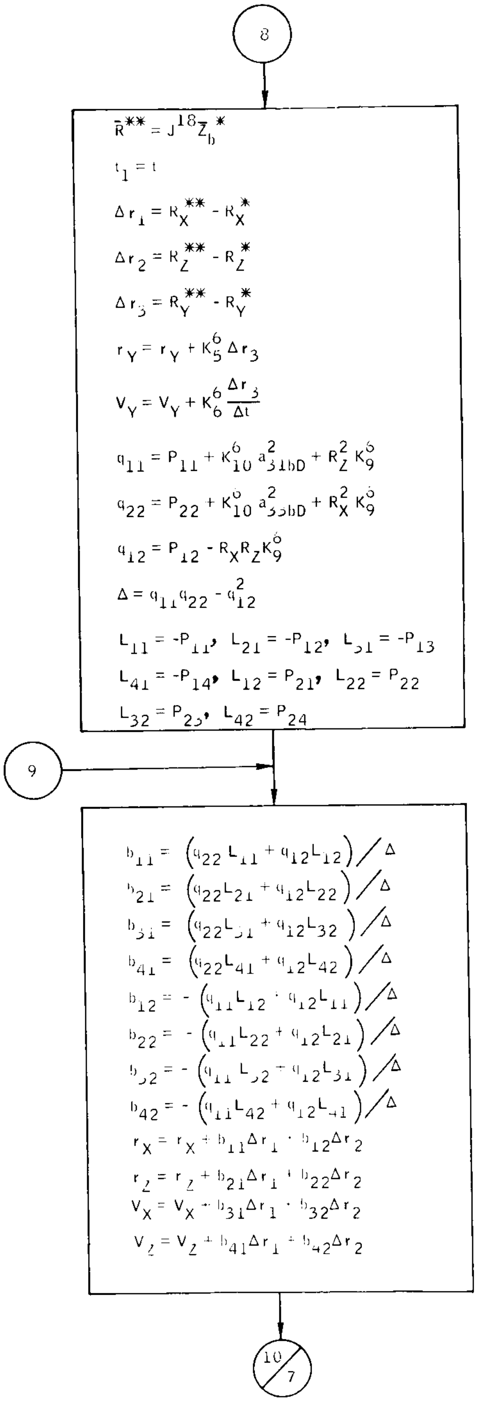
Page 4 of 8

Figure B-38.- Continued.


6
KA(1) = 0
KA(2) = 0
INITRR = 1
CALL AGSKEP
INITIALIZE AGSKEP
10A
7
7
NDEL = 0
NO
8
6
YES
$\Delta r_1 = J^{17} - \dot{R}^*$
$J^{17} = 0$
$q_{12} = 0$
$q_{22} = 1$
$M_{11} = \left(\dot{R}a_{31bD} - \dot{R}_X\right) / R$
$M_{13} = -a_{31bD}$
$M_{14} = -a_{33bD}$
$M_{12} = \left(\dot{R}a_{33bD} - \dot{R}_Z\right) / R$
$L_{12} = L_{22} = L_{32} = L_{42} = 0$
$L_{11} = P_{11}M_{11} + P_{21}M_{12} + P_{13}M_{13} + P_{14}M_{14}$
$L_{21} = P_{12}M_{11} + P_{22}M_{12} + P_{23}M_{13} + P_{24}M_{14}$
$L_{31} = P_{13}M_{11} + P_{23}M_{12} + P_{33}M_{13} + P_{34}M_{14}$
$L_{41} = P_{14}M_{11} + P_{24}M_{12} + P_{43}M_{13} + P_{44}M_{14}$
$q_{11} = \Delta = M_{11}L_{11} + M_{12}L_{21} + M_{13}L_{31} + M_{14}L_{41} + K_8^6$
9
6

Page 5 of 8

Figure B-38.- Continued.



$$\bar{R}^{**} = J^{18}\bar{Z}_b^{*}$$

$$t_1 = t$$

$$\Delta r_1 = R_X^{**} - R_X^{*}$$

$$\Delta r_2 = R_Z^{**} - R_Z^{*}$$

$$\Delta r_3 = R_Y^{**} - R_Y^{*}$$

$$r_Y = r_Y + K_5^6 \Delta r_3$$

$$V_Y = V_Y + K_6^6 \frac{\Delta r_3}{\Delta t}$$

$$q_{11} = P_{11} + K_{10}^6 a_{31bD}^2 + R_Z^2 K_9^6$$

$$q_{22} = P_{22} + K_{10}^6 a_{33bD}^2 + R_X^2 K_9^6$$

$$q_{12} = P_{12} - R_X R_Z K_9^6$$

$$\Delta = q_{11} q_{22} - q_{12}^2$$

$$L_{11} = -P_{11}, \ L_{21} = -P_{12}, \ L_{31} = -P_{13}$$

$$L_{41} = -P_{14}, \ L_{12} = P_{21}, \ L_{22} = P_{22}$$

$$L_{32} = P_{23}, \ L_{42} = P_{24}$$

$$b_{11} = \left(q_{22} L_{11} + q_{12} L_{12}\right) \big/ \Delta$$

$$b_{21} = \left(q_{22} L_{21} + q_{12} L_{22}\right) \big/ \Delta$$

$$b_{31} = \left(q_{22} L_{31} + q_{12} L_{32}\right) \big/ \Delta$$

$$b_{41} = \left(q_{22} L_{41} + q_{12} L_{42}\right) \big/ \Delta$$

$$b_{12} = -\left(q_{11} L_{12} + q_{12} L_{11}\right) \big/ \Delta$$

$$b_{22} = -\left(q_{11} L_{22} + q_{12} L_{21}\right) \big/ \Delta$$

$$b_{32} = -\left(q_{11} L_{32} + q_{12} L_{31}\right) \big/ \Delta$$

$$b_{42} = -\left(q_{11} L_{42} + q_{12} L_{41}\right) \big/ \Delta$$

$$r_X = r_X + b_{11} \Delta r_1 + b_{12} \Delta r_2$$

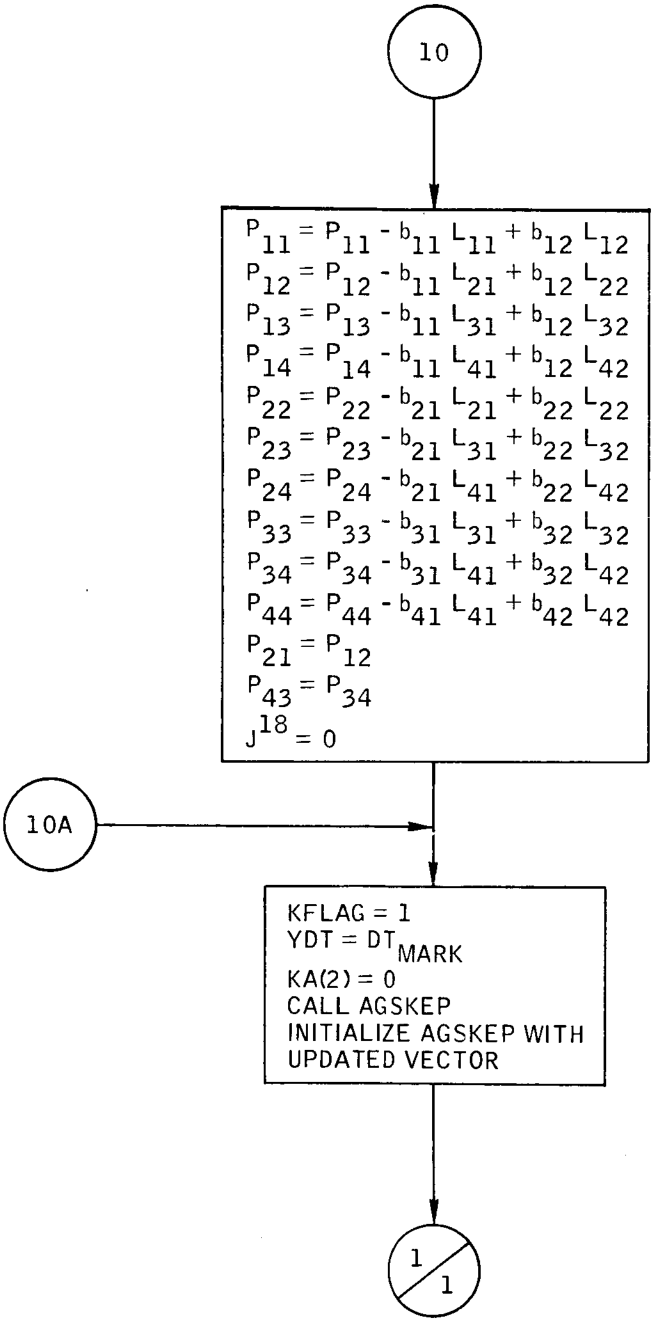
$$r_Z = r_Z + b_{21} \Delta r_1 + b_{22} \Delta r_2$$

$$V_X = V_X + b_{31} \Delta r_1 + b_{32} \Delta r_2$$

$$V_Z = V_Z + b_{41} \Delta r_1 + b_{42} \Delta r_2$$

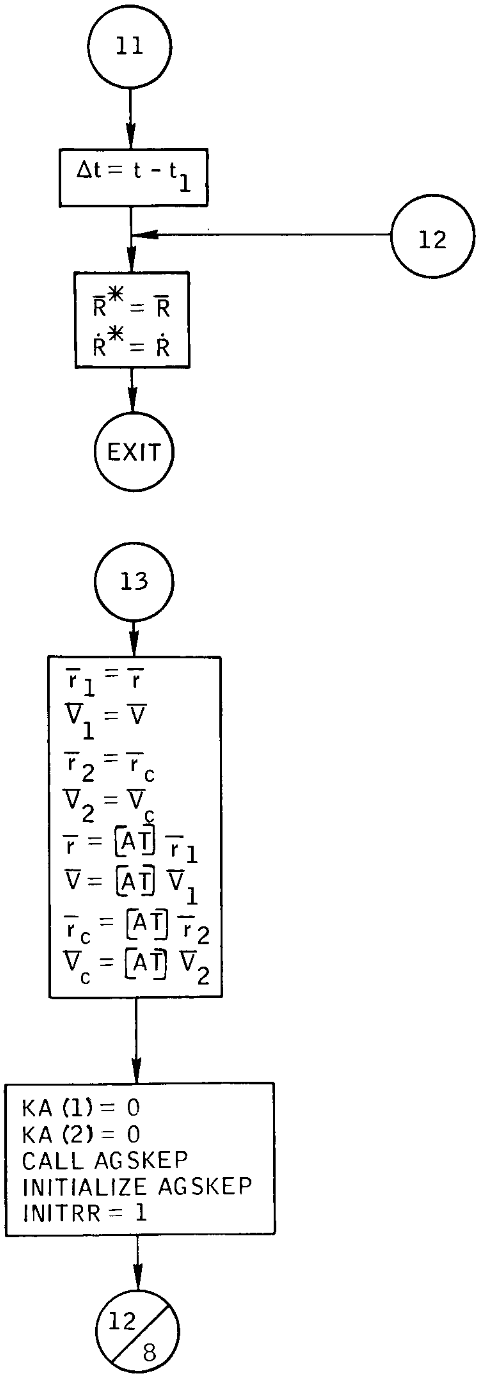
Page 3 of 6

Figure B-38.- Continued.


10
$P_{11} = P_{11} - b_{11} L_{11} + b_{12} L_{12}$
$P_{12} = P_{12} - b_{11} L_{21} + b_{12} L_{22}$
$P_{13} = P_{13} - b_{11} L_{31} + b_{12} L_{32}$
$P_{14} = P_{14} - b_{11} L_{41} + b_{12} L_{42}$
$P_{22} = P_{22} - b_{21} L_{21} + b_{22} L_{22}$
$P_{23} = P_{23} - b_{21} L_{31} + b_{22} L_{32}$
$P_{24} = P_{24} - b_{21} L_{41} + b_{22} L_{42}$
$P_{33} = P_{33} - b_{31} L_{31} + b_{32} L_{32}$
$P_{34} = P_{34} - b_{31} L_{41} + b_{32} L_{42}$
$P_{44} = P_{44} - b_{41} L_{41} + b_{42} L_{42}$
$P_{21} = P_{12}$
$P_{43} = P_{34}$
$J^{18} = 0$
10A
KFLAG = 1
YDT = DT_{MARK}
KA(2) = 0
CALL AGSKEP
INITIALIZE AGSKEP WITH
UPDATED VECTOR
1
1

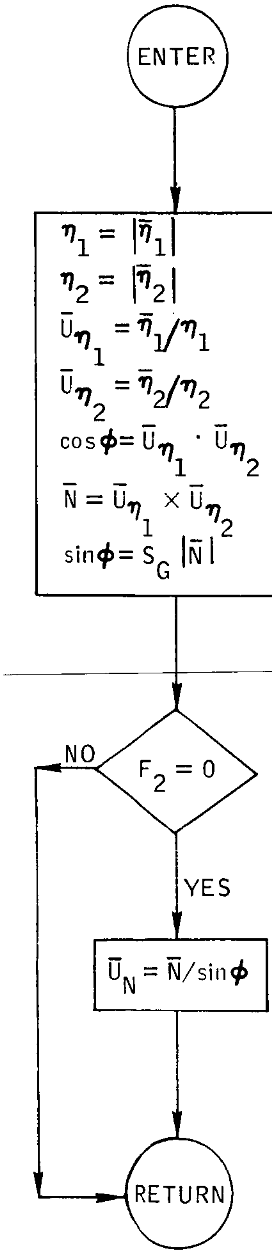
Page 7 of 8

Figure B-38.- Continued.



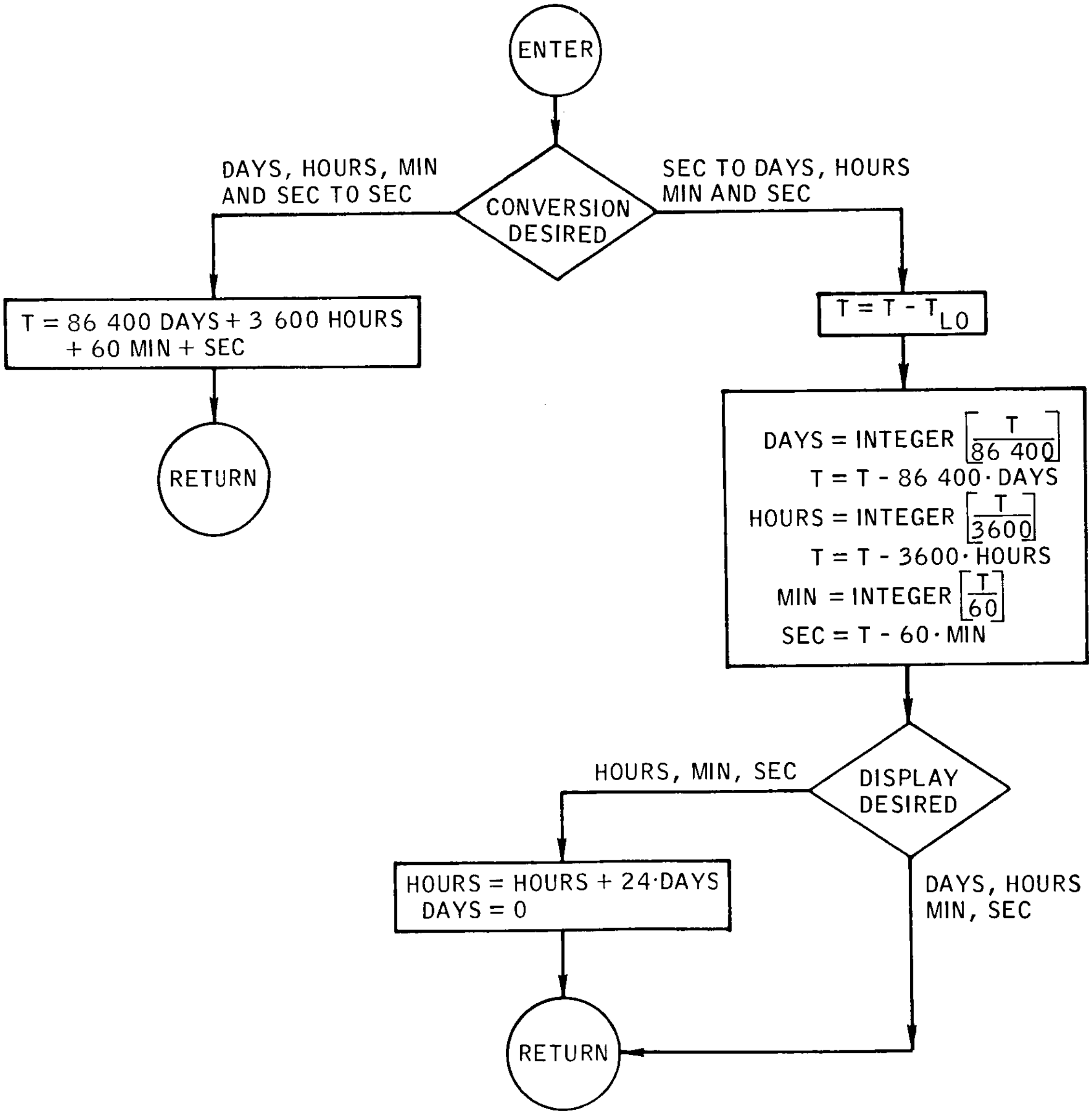
Page 8 of 8

Figure B-38.- Concluded.


ENTER
$\eta_1 = |\bar{\eta}_1|$
$\eta_2 = |\bar{\eta}_2|$
$\bar{U}_{\eta_1} = \bar{\eta}_1/\eta_1$
$\bar{U}_{\eta_2} = \bar{\eta}_2/\eta_2$
$\cos\phi = \bar{U}_{\eta_1} \cdot \bar{U}_{\eta_2}$
$\bar{N} = \bar{U}_{\eta_1} \times \bar{U}_{\eta_2}$
$\sin\phi = S_G\,|\bar{N}|$
$F_2 = 0$
NO
YES
$\bar{U}_N = \bar{N}/\sin\phi$
RETURN

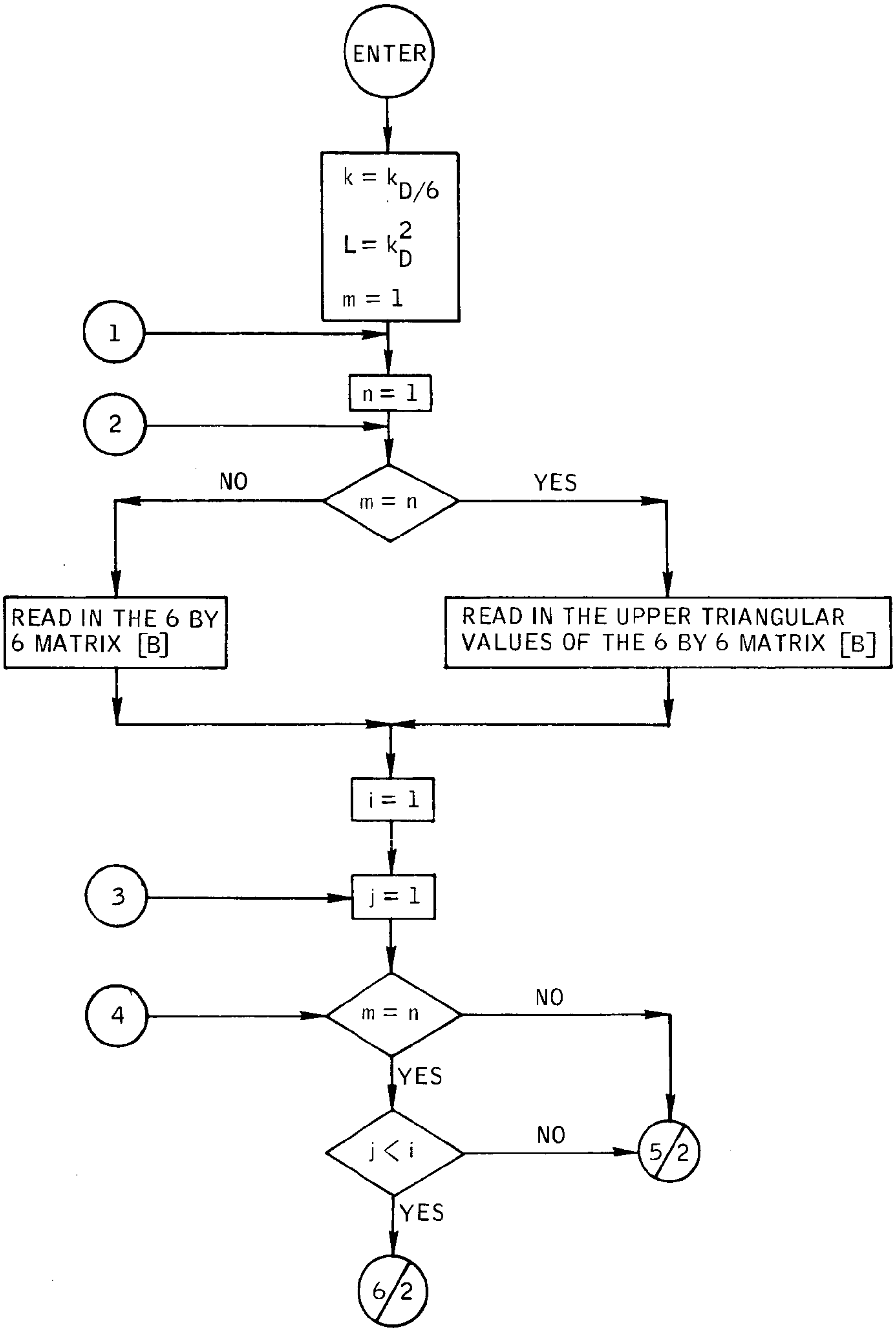
Page 1 of 1

Figure B-39.- Flow chart of subroutine GEOMET.


ENTER
CONVERSION DESIRED
DAYS, HOURS, MIN AND SEC TO SEC
SEC TO DAYS, HOURS MIN AND SEC
T = 86 400 DAYS + 3 600 HOURS + 60 MIN + SEC
RETURN
T = T - T_{LO}
DAYS = INTEGER $\left[\frac{T}{86\ 400}\right]$
T = T - 86 400 · DAYS
HOURS = INTEGER $\left[\frac{T}{3600}\right]$
T = T - 3600 · HOURS
MIN = INTEGER $\left[\frac{T}{60}\right]$
SEC = T - 60 · MIN
DISPLAY DESIRED
HOURS, MIN, SEC
DAYS, HOURS MIN, SEC
HOURS = HOURS + 24·DAYS
DAYS = 0
RETURN

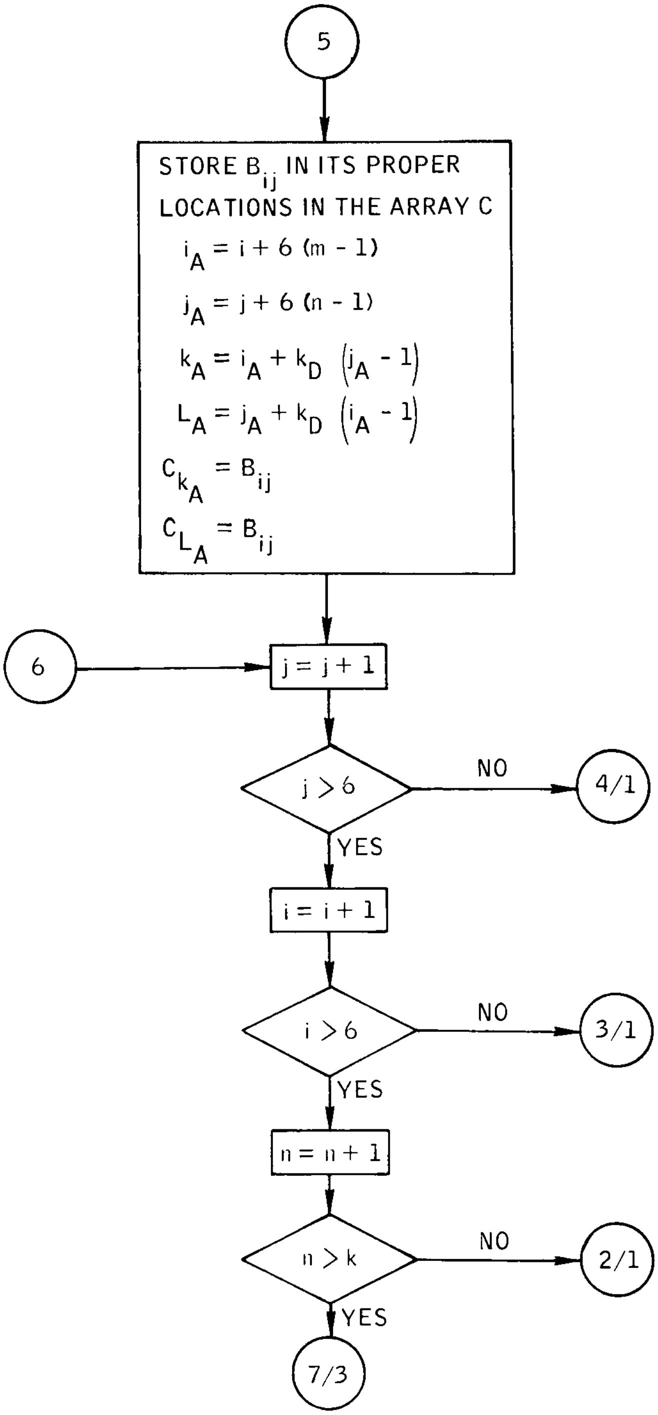
Page 1 of 1

Figure B-40.- Flow chart of subroutine GMTIME.



Page 1 of 3

Figure B-41.- Flow chart of subroutine INCOV.



Page 2 of 3

Figure B-41.- Continued.

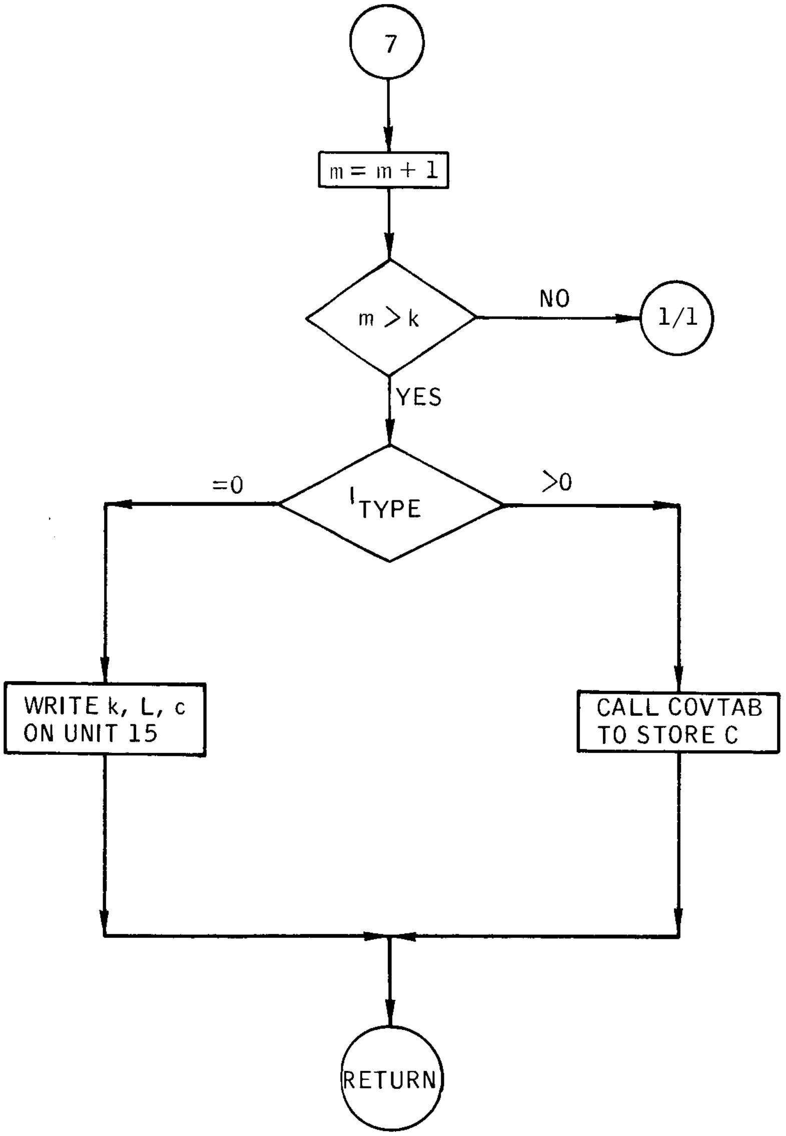
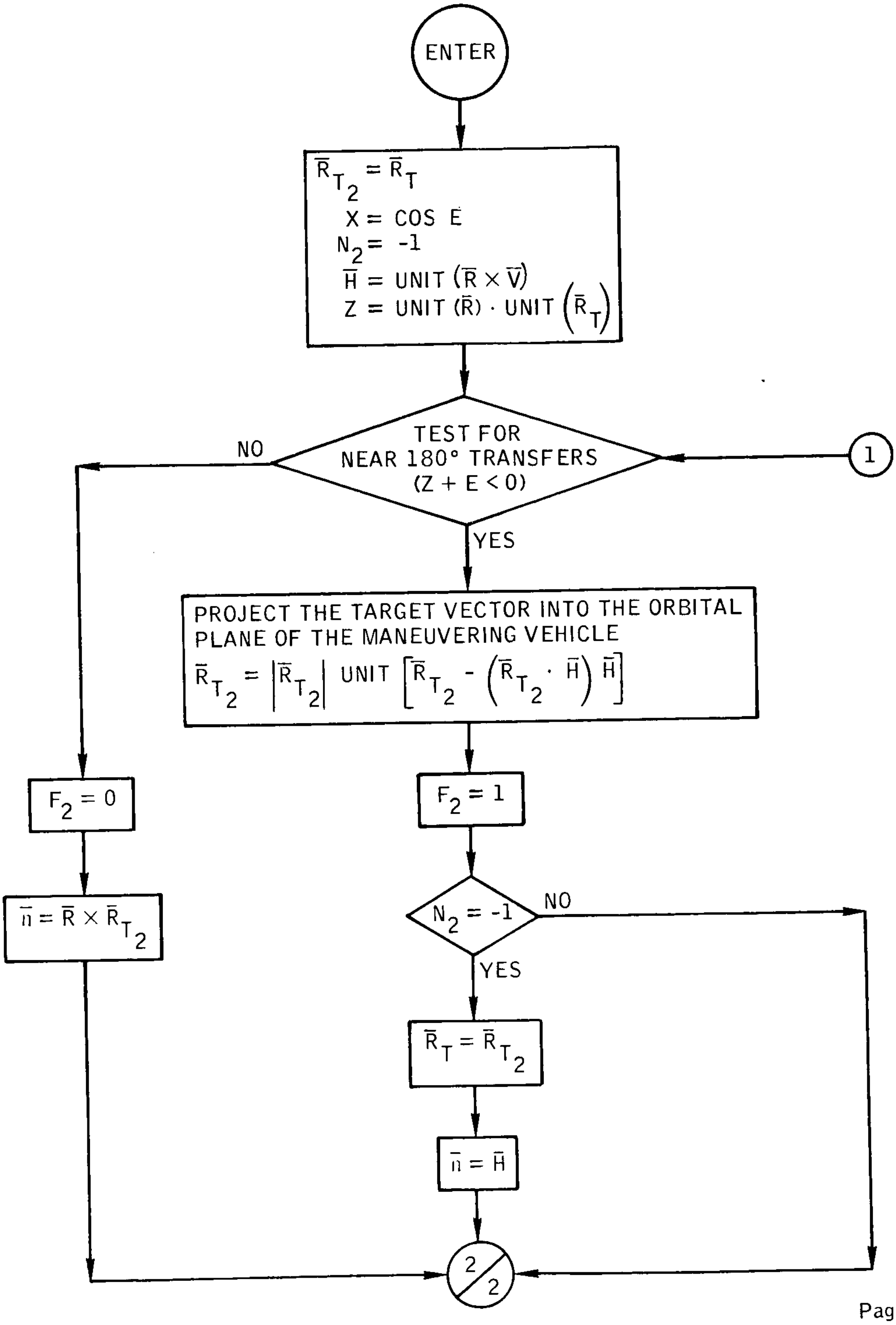
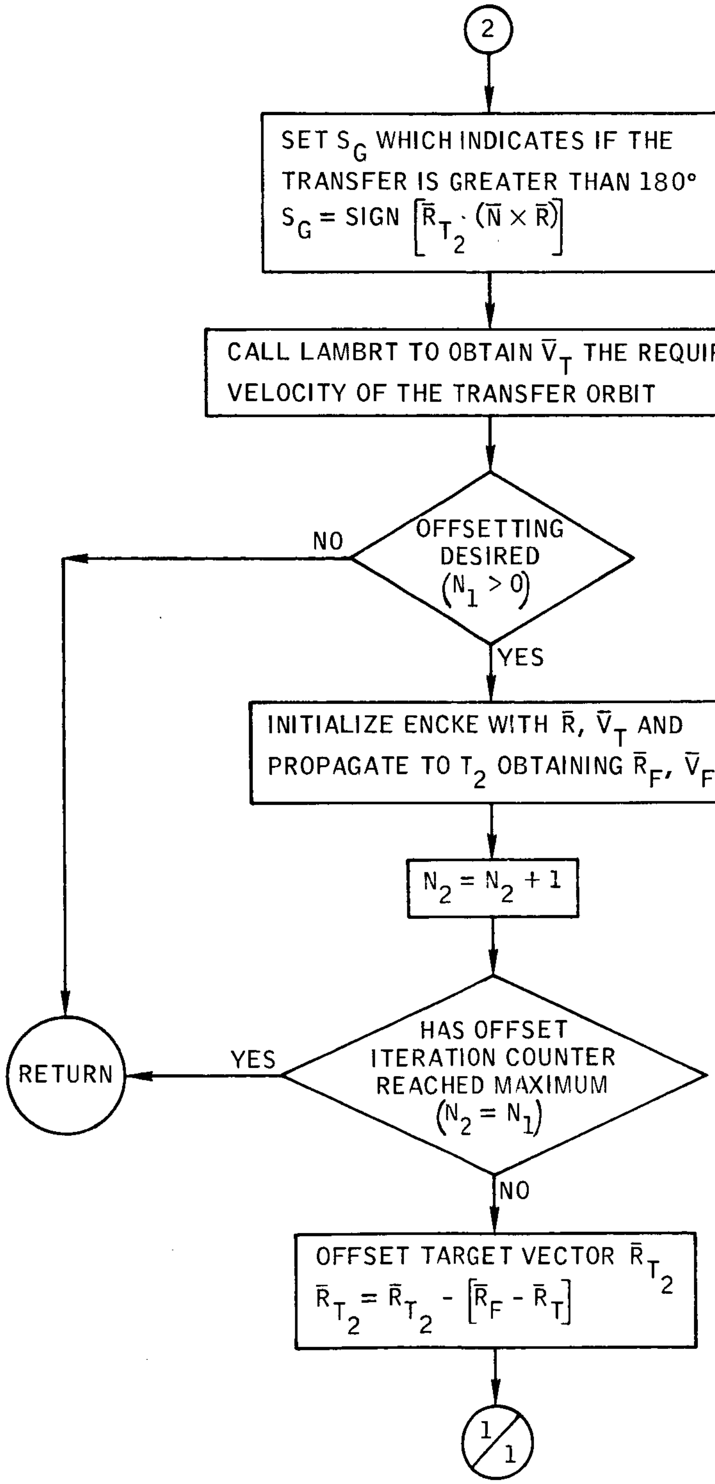


Figure B-41.- Concluded.


ENTER
$\bar{R}_{T_2} = \bar{R}_T$
$X = \cos E$
$N_2 = -1$
$\bar{H} = \text{UNIT}(\bar{R} \times \bar{V})$
$Z = \text{UNIT}(\bar{R}) \cdot \text{UNIT}(\bar{R}_T)$
TEST FOR NEAR 180° TRANSFERS $(Z + E < 0)$
NO
YES
1
PROJECT THE TARGET VECTOR INTO THE ORBITAL PLANE OF THE MANEUVERING VEHICLE
$\bar{R}_{T_2} = |\bar{R}_{T_2}| \text{ UNIT } \left[\bar{R}_{T_2} - (\bar{R}_{T_2} \cdot \bar{H})\bar{H}\right]$
$F_2 = 0$
$F_2 = 1$
$\bar{n} = \bar{R} \times \bar{R}_{T_2}$
$N_2 = -1$
NO
YES
$\bar{R}_T = \bar{R}_{T_2}$
$\bar{n} = \bar{H}$
2/2

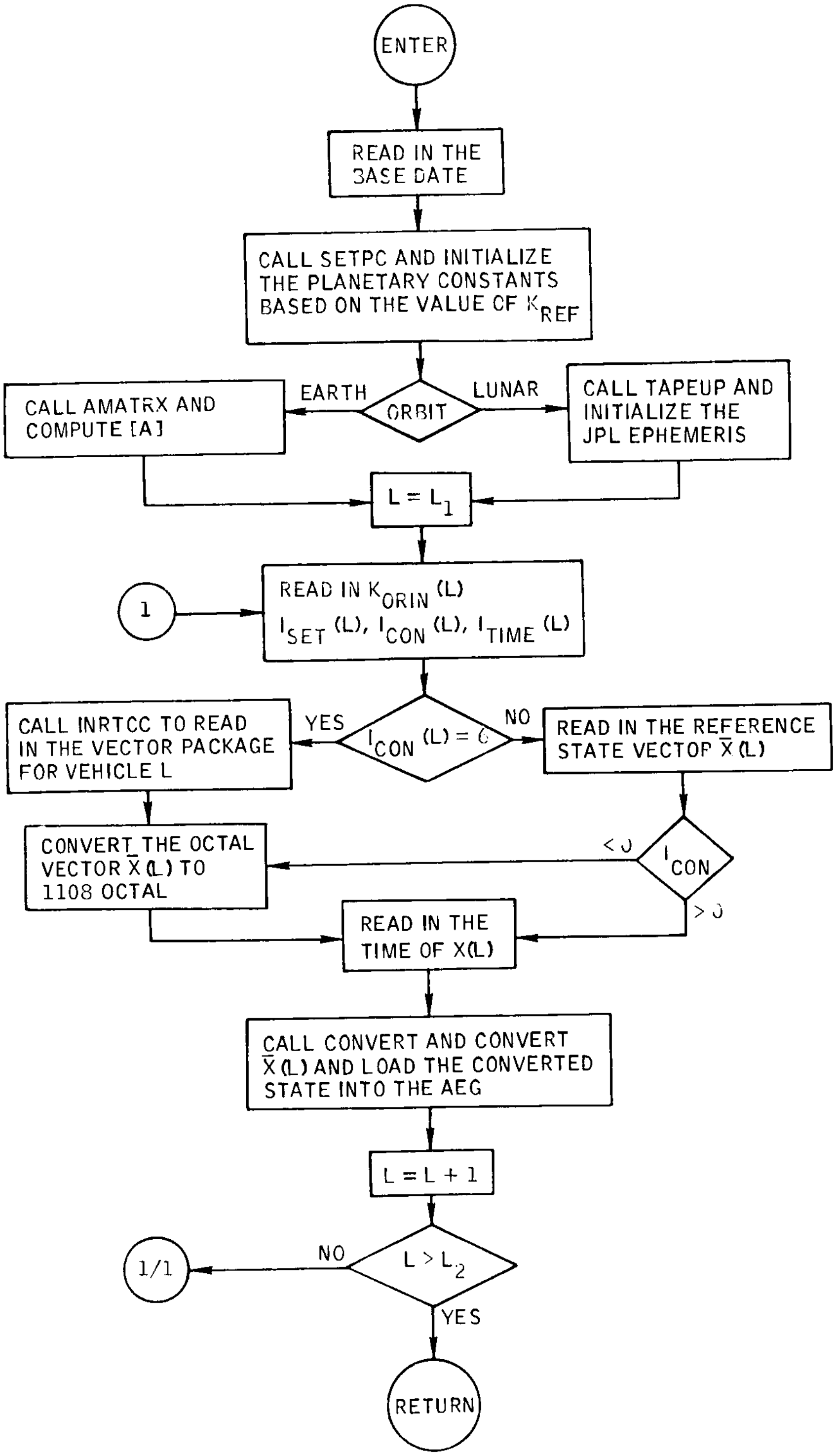
Page 1 of 2

Figure B-42.- Flow chart of subroutine INITV.


2
SET S_G WHICH INDICATES IF THE TRANSFER IS GREATER THAN 180°
$S_G = \text{SIGN}\left[\bar{R}_{T_2} \cdot (\bar{N} \times \bar{R})\right]$
CALL LAMBRT TO OBTAIN $\bar{V}_T$ THE REQUIRED VELOCITY OF THE TRANSFER ORBIT
OFFSETTING DESIRED $(N_1 > 0)$
NO
YES
INITIALIZE ENCKE WITH $\bar{R}$, $\bar{V}_T$ AND PROPAGATE TO T_2 OBTAINING $\bar{R}_F$, $\bar{V}_F$
$N_2 = N_2 + 1$
HAS OFFSET ITERATION COUNTER REACHED MAXIMUM $(N_2 = N_1)$
YES
RETURN
NO
OFFSET TARGET VECTOR $\bar{R}_{T_2}$
$\bar{R}_{T_2} = \bar{R}_{T_2} - \left[\bar{R}_F - \bar{R}_T\right]$
1/1

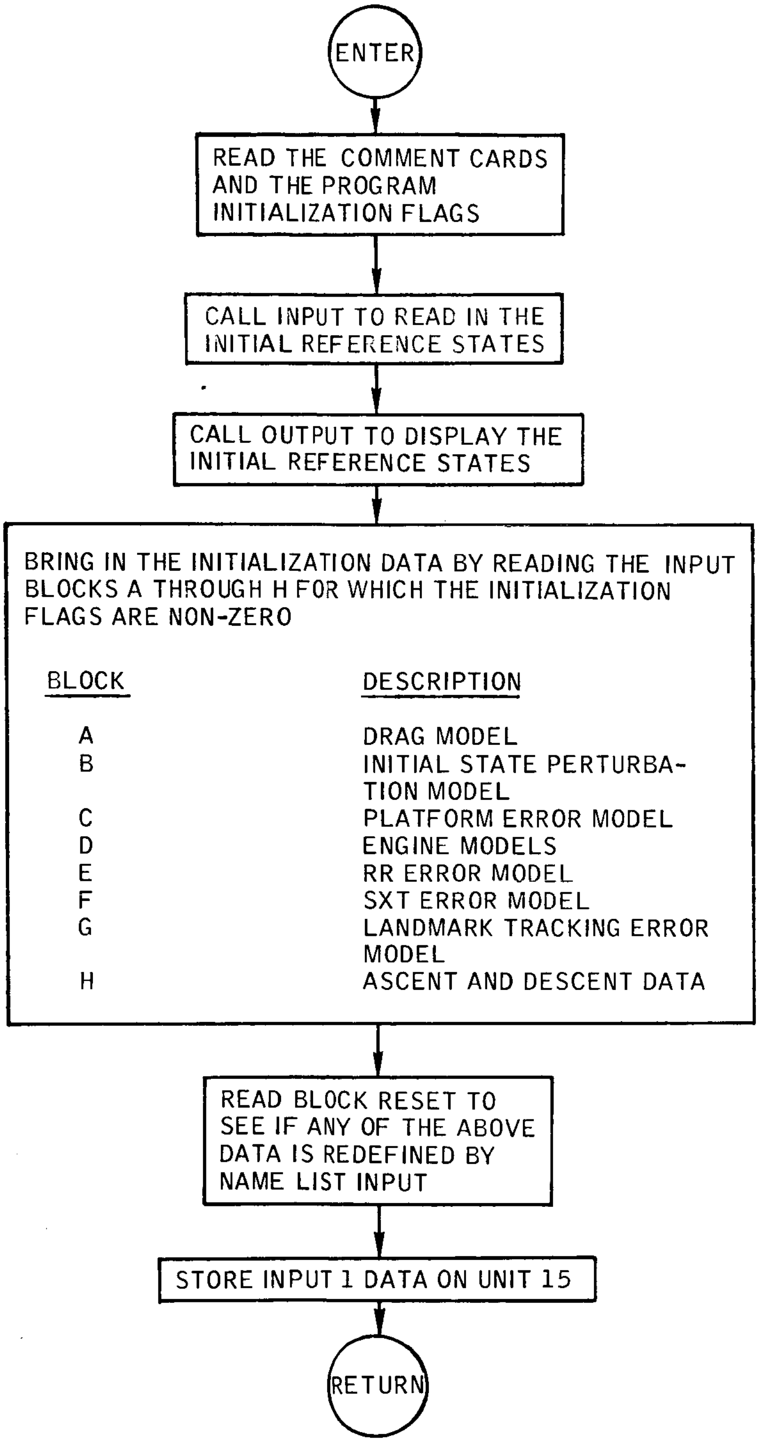
Page 2 of 2

Figure B-42.- Concluded.


ENTER
READ IN THE 3ASE DATE
CALL SETPC AND INITIALIZE THE PLANETARY CONSTANTS BASED ON THE VALUE CF K_{REF}
EARTH
ORBIT
LUNAR
CALL AMATRX AND COMPUTE [A]
CALL TAPEUP AND INITIALIZE THE JPL EPHEMERIS
$L = L_1$
1
READ IN K_{ORIN} (L)
I_{SET} (L), I_{CON} (L), I_{TIME} (L)
YES
I_{CON} (L) = 0
NO
CALL INRTCC TO READ IN THE VECTOR PACKAGE FOR VEHICLE L
READ IN THE REFERENCE STATE VECTOF $\overline{X}$ (L)
CONVERT THE OCTAL VECTOR $\overline{X}$ (L) TO 1108 OCTAL
< 0
I_{CON}
> 0
READ IN THE TIME OF X(L)
CALL CONVERT AND CONVERT $\overline{X}$ (L) AND LOAD THE CONVERTED STATE INTO THE AEG
L = L + 1
NO
$L > L_2$
1/1
YES
RETURN

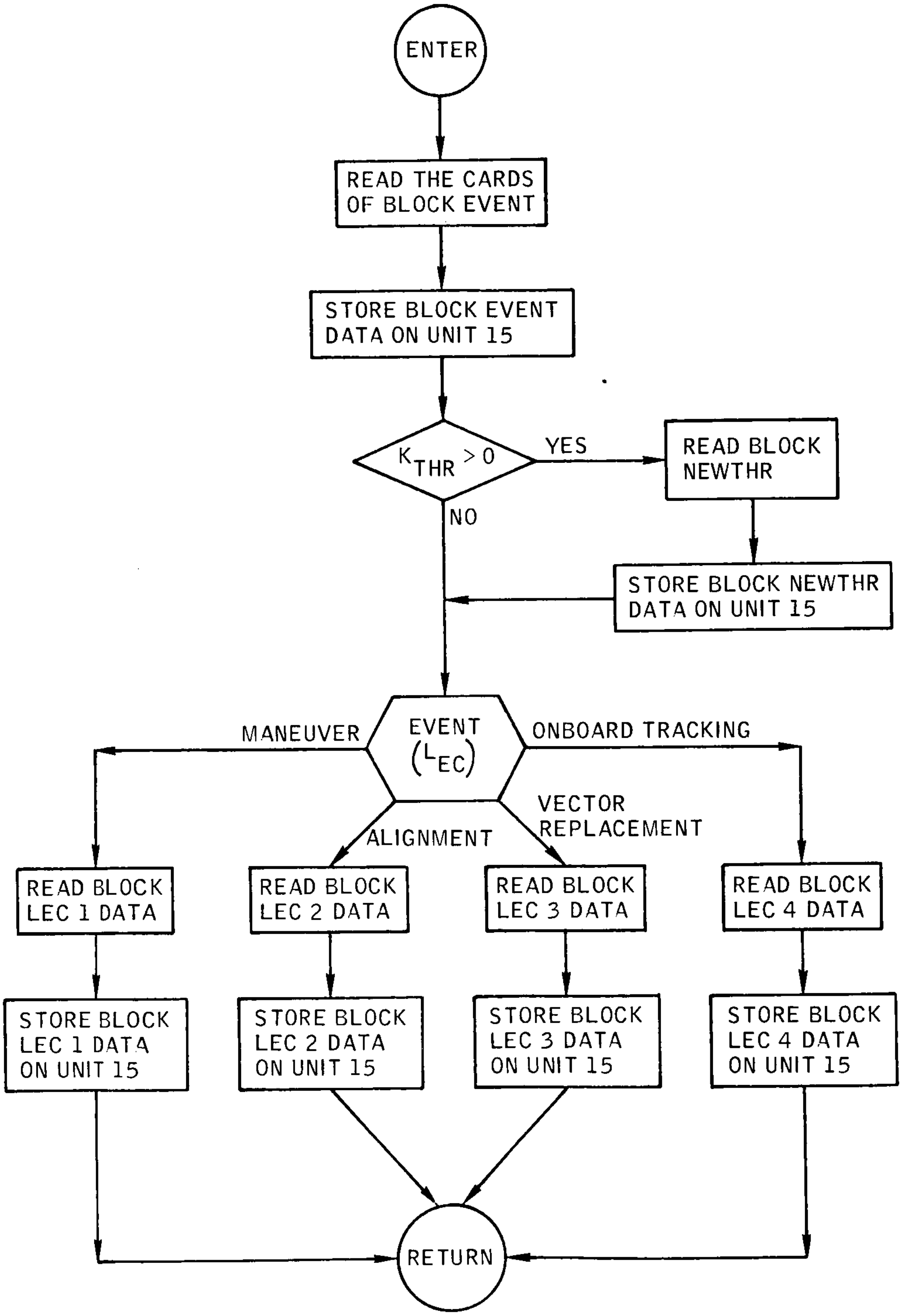
Page 1 of 1

Figure B-43.- Flow chart of subroutine INPUT.


ENTER
READ THE COMMENT CARDS AND THE PROGRAM INITIALIZATION FLAGS
CALL INPUT TO READ IN THE INITIAL REFERENCE STATES
CALL OUTPUT TO DISPLAY THE INITIAL REFERENCE STATES
BRING IN THE INITIALIZATION DATA BY READING THE INPUT BLOCKS A THROUGH H FOR WHICH THE INITIALIZATION FLAGS ARE NON-ZERO
BLOCK
DESCRIPTION
A
DRAG MODEL
B
INITIAL STATE PERTURBATION MODEL
C
PLATFORM ERROR MODEL
D
ENGINE MODELS
E
RR ERROR MODEL
F
SXT ERROR MODEL
G
LANDMARK TRACKING ERROR MODEL
H
ASCENT AND DESCENT DATA
READ BLOCK RESET TO SEE IF ANY OF THE ABOVE DATA IS REDEFINED BY NAME LIST INPUT
STORE INPUT 1 DATA ON UNIT 15
RETURN

Page 1 of 1

Figure B-44.- Flow chart of subroutine INPUT 1.


ENTER
READ THE CARDS OF BLOCK EVENT
STORE BLOCK EVENT DATA ON UNIT 15
$K_{THR} > 0$
YES
NO
READ BLOCK NEWTHR
STORE BLOCK NEWTHR DATA ON UNIT 15
EVENT (L_{EC})
MANEUVER
ALIGNMENT
VECTOR REPLACEMENT
ONBOARD TRACKING
READ BLOCK LEC 1 DATA
READ BLOCK LEC 2 DATA
READ BLOCK LEC 3 DATA
READ BLOCK LEC 4 DATA
STORE BLOCK LEC 1 DATA ON UNIT 15
STORE BLOCK LEC 2 DATA ON UNIT 15
STORE BLOCK LEC 3 DATA ON UNIT 15
STORE BLOCK LEC 4 DATA ON UNIT 15
RETURN

Page 1 of 1

Figure B-45.- Flow chart of subroutine INPUT 2.

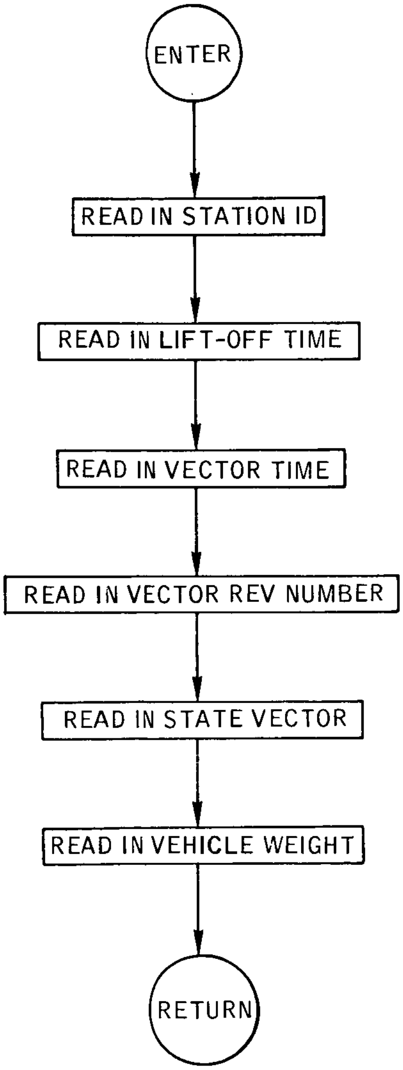


Figure B-46.- Flow chart of subroutine INRTCC.

Page 1 of 1

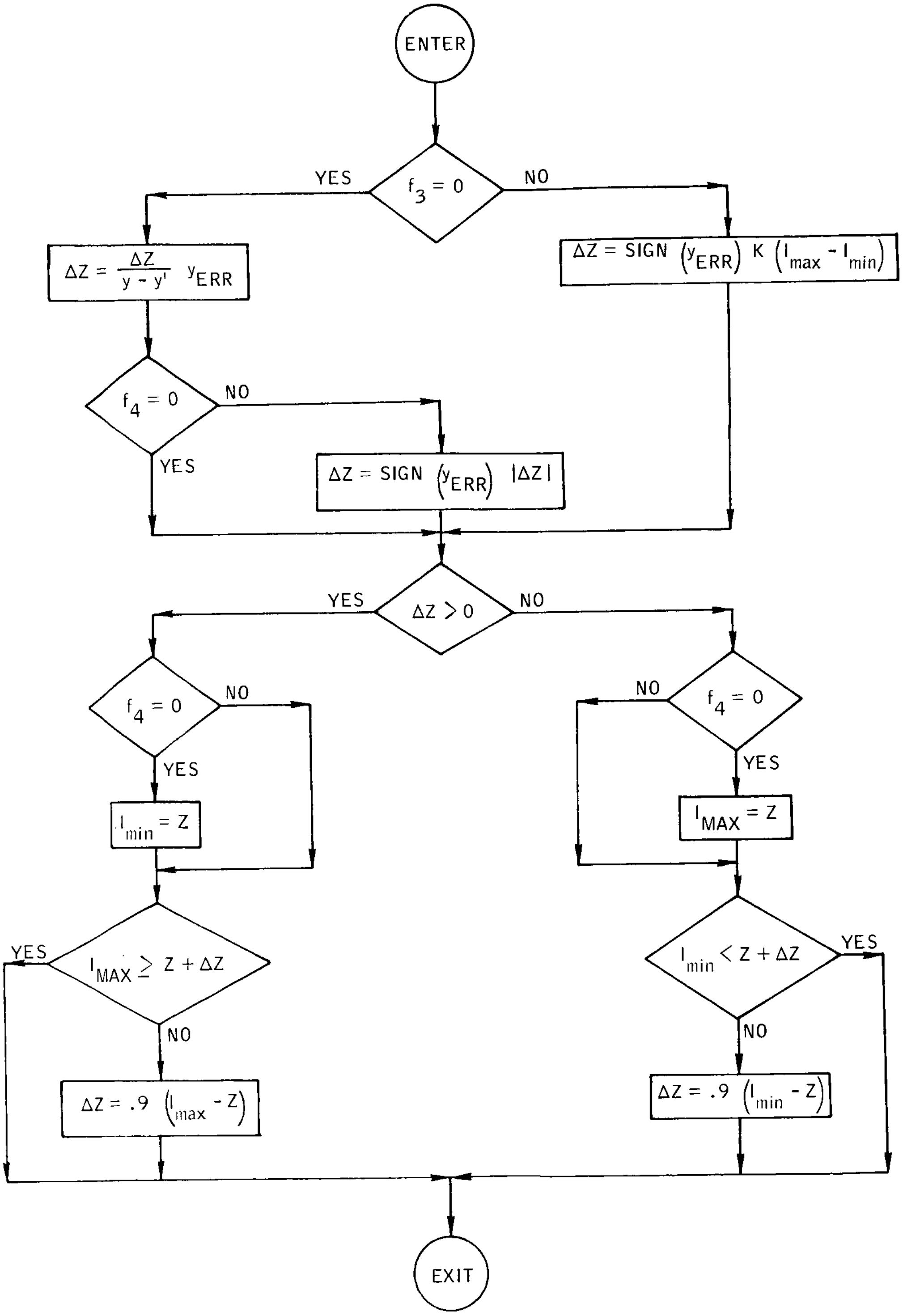
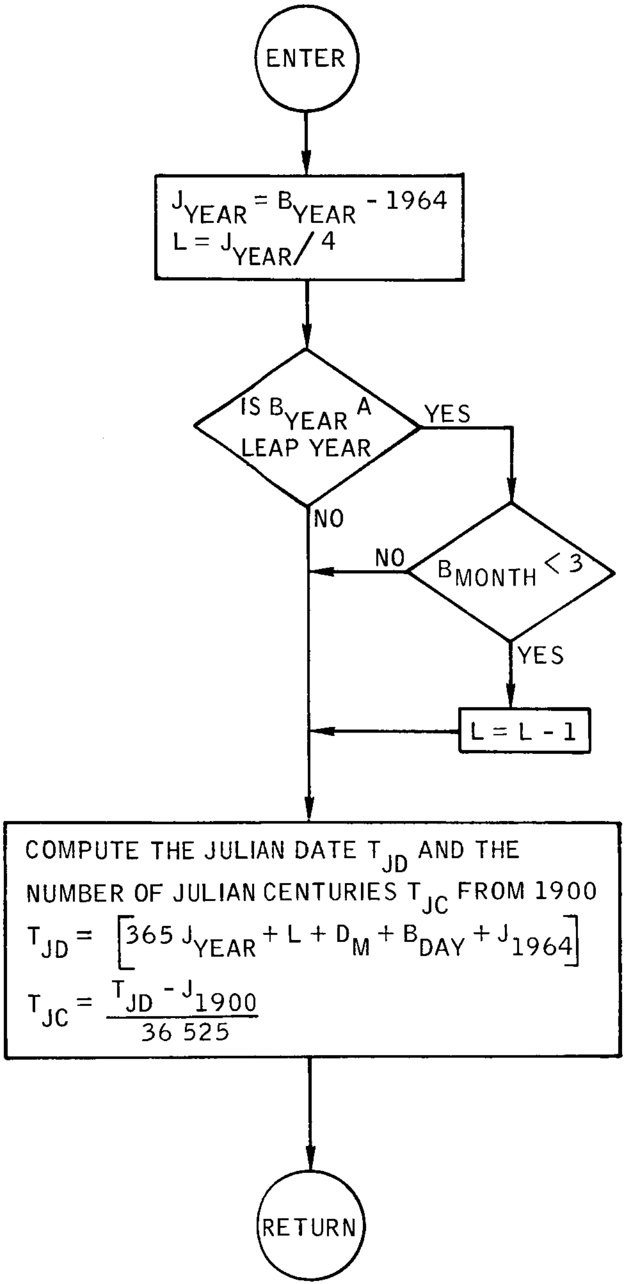

ENTER
$f_3 = 0$
YES
NO
$\Delta Z = \frac{\Delta Z}{y - y'}\ y_{ERR}$
$\Delta Z = \text{SIGN}\left(y_{ERR}\right) K \left(I_{max} - I_{min}\right)$
$f_4 = 0$
NO
YES
$\Delta Z = \text{SIGN}\left(y_{ERR}\right) |\Delta Z|$
$\Delta Z > 0$
YES
NO
$f_4 = 0$
NO
YES
$I_{min} = Z$
$f_4 = 0$
NO
YES
$I_{MAX} = Z$
$I_{MAX} \geq Z + \Delta Z$
YES
NO
$\Delta Z = .9\left(I_{max} - Z\right)$
$I_{min} < Z + \Delta Z$
YES
NO
$\Delta Z = .9\left(I_{min} - Z\right)$
EXIT

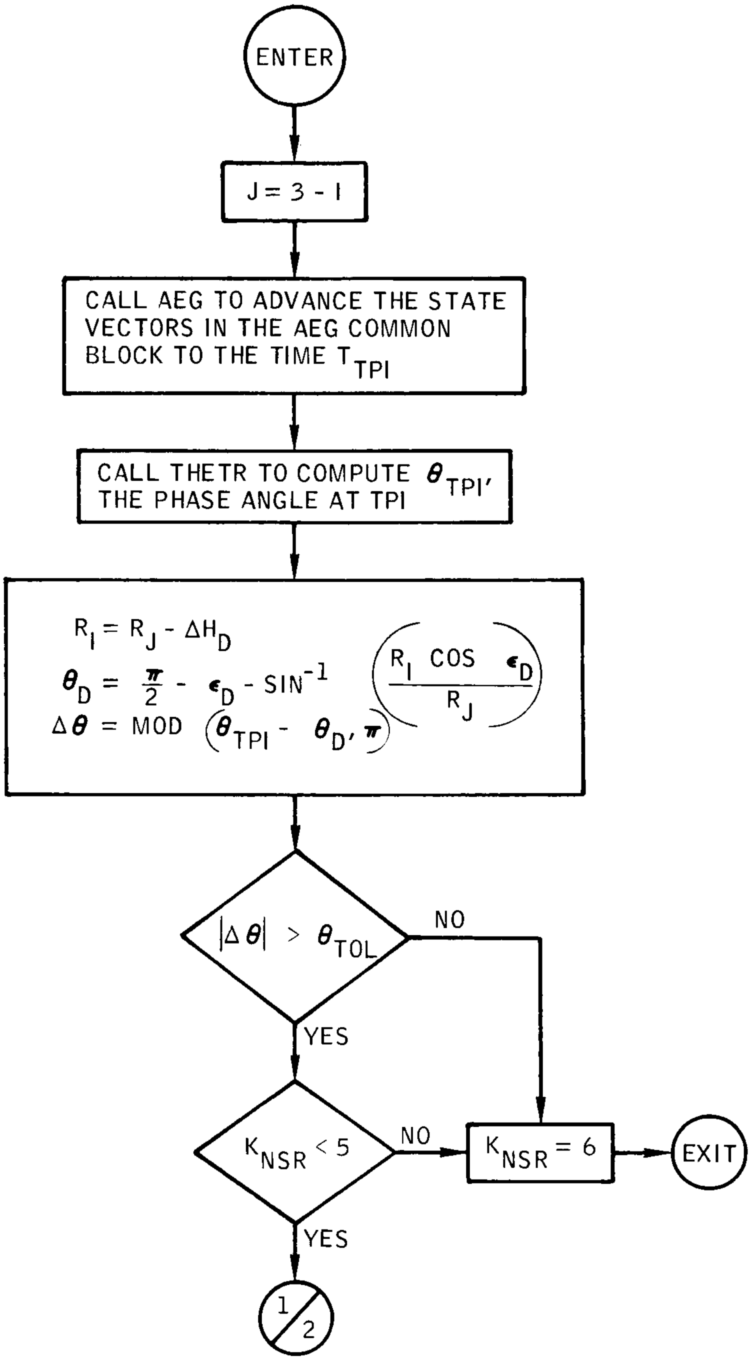
Figure B-47.- Flow chart of subroutine ITRATE.

Page 1 of 1


ENTER
$J_{YEAR} = B_{YEAR} - 1964$
$L = J_{YEAR}/4$
IS B_{YEAR} A LEAP YEAR
YES
NO
NO
$B_{MONTH} < 3$
YES
$L = L - 1$
COMPUTE THE JULIAN DATE T_{JD} AND THE NUMBER OF JULIAN CENTURIES T_{JC} FROM 1900
$T_{JD} = \left[365\ J_{YEAR} + L + D_M + B_{DAY} + J_{1964}\right]$
$T_{JC} = \frac{T_{JD} - J_{1900}}{36\ 525}$
RETURN

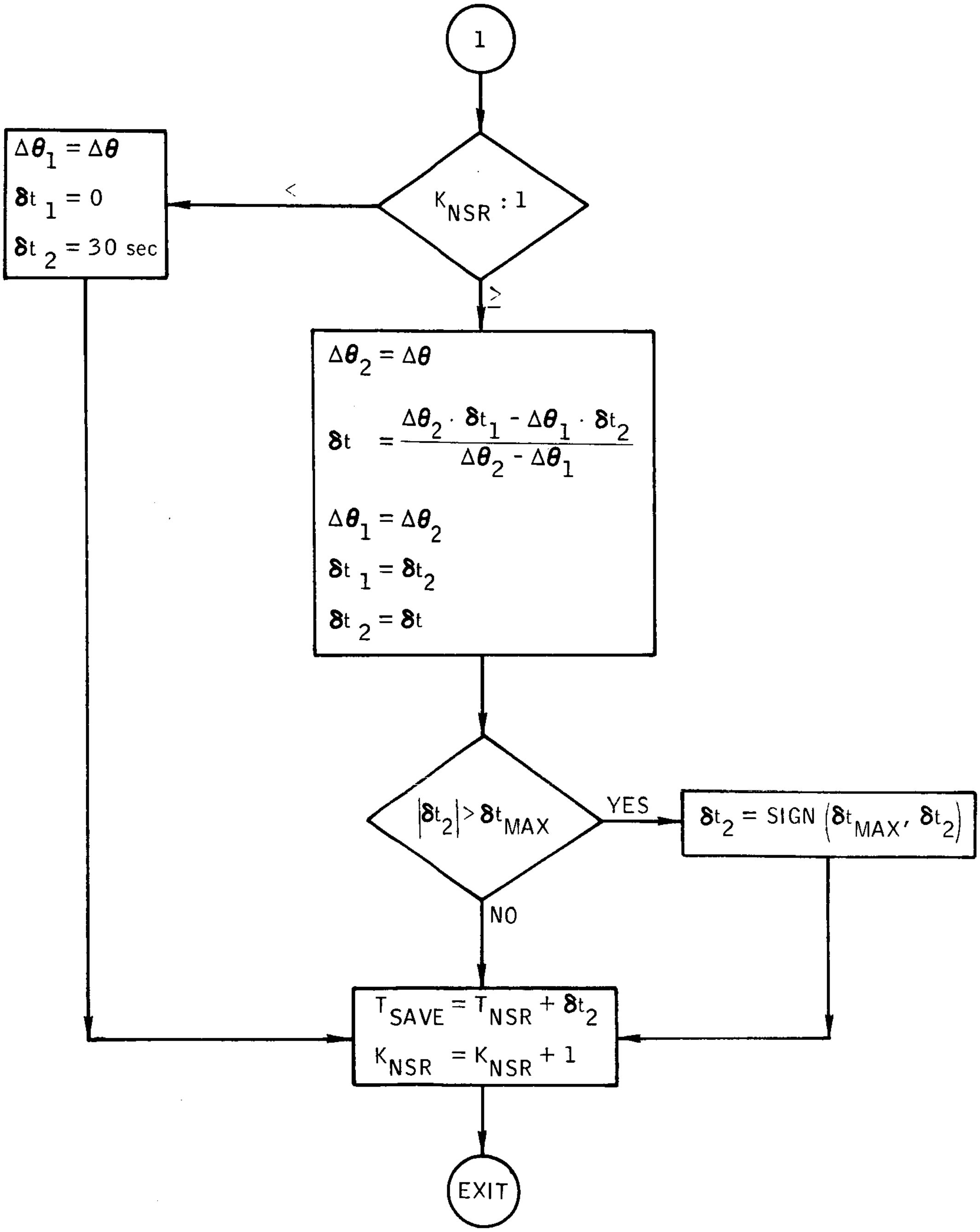
Page 1 of 1

Figure B-48.- Flow chart of subroutine JDATE.


ENTER
J = 3 - I
CALL AEG TO ADVANCE THE STATE VECTORS IN THE AEG COMMON BLOCK TO THE TIME T_{TPI}
CALL THETR TO COMPUTE θ_{TPI}, THE PHASE ANGLE AT TPI
$R_I = R_J - \Delta H_D$
$\theta_D = \frac{\pi}{2} - \epsilon_D - \mathrm{SIN}^{-1} \left(\frac{R_I \ \mathrm{COS} \ \epsilon_D}{R_J} \right)$
$\Delta\theta = \mathrm{MOD} \left(\theta_{TPI} - \theta_D, \pi \right)$
$|\Delta\theta| > \theta_{TOL}$
NO
YES
$K_{NSR} < 5$
NO
$K_{NSR} = 6$
EXIT
YES
1/2

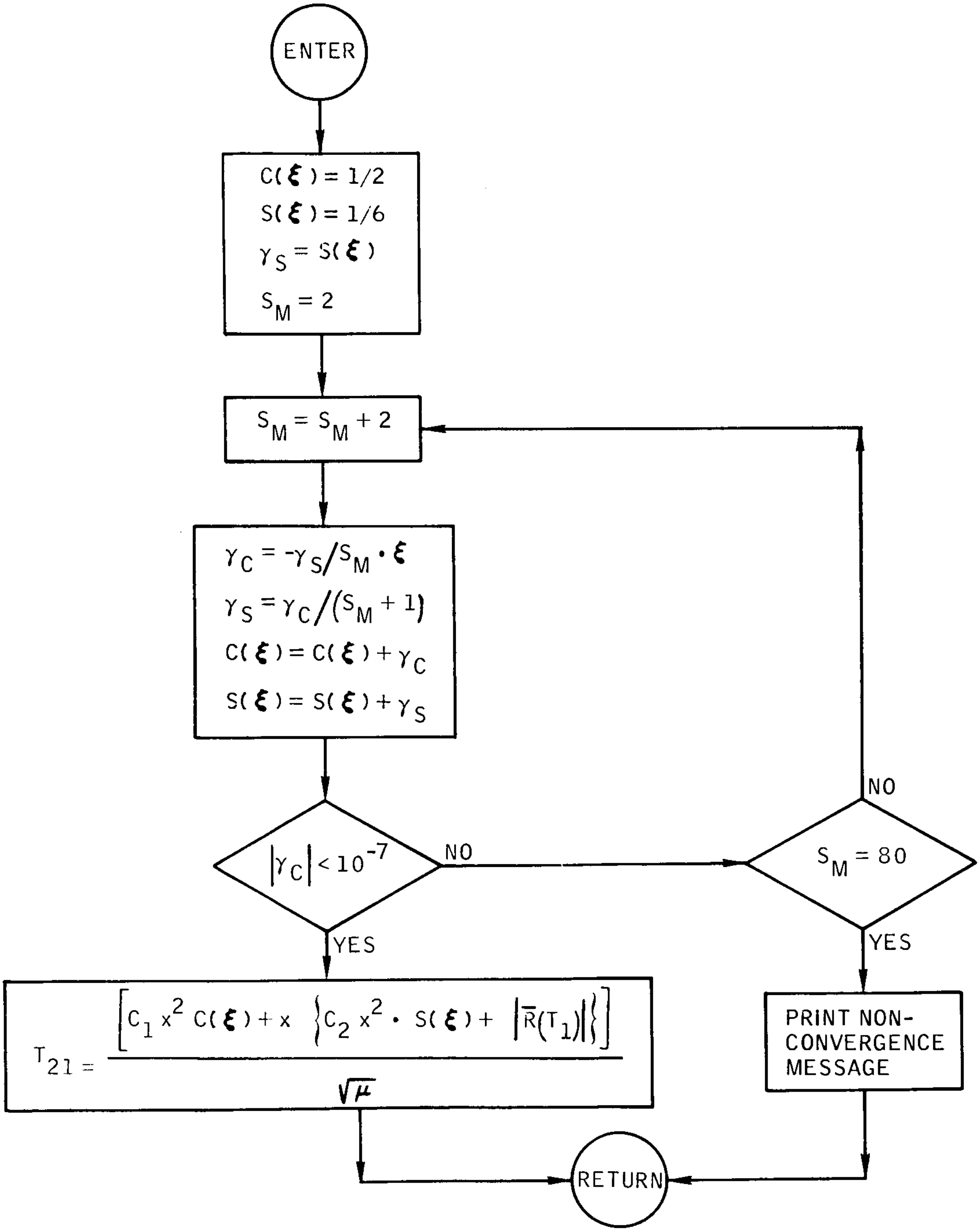
Page 1 of 2

Figure B-49.- Flow chart of subroutine JOCDH.


1
$K_{NSR} : 1$
<
$\Delta\theta_1 = \Delta\theta$
$\delta t_1 = 0$
$\delta t_2 = 30$ sec
≥
$\Delta\theta_2 = \Delta\theta$
$\delta t = \frac{\Delta\theta_2 \cdot \delta t_1 - \Delta\theta_1 \cdot \delta t_2}{\Delta\theta_2 - \Delta\theta_1}$
$\Delta\theta_1 = \Delta\theta_2$
$\delta t_1 = \delta t_2$
$\delta t_2 = \delta t$
$|\delta t_2| > \delta t_{MAX}$
YES
$\delta t_2 = \mathrm{SIGN}(\delta t_{MAX}, \delta t_2)$
NO
$T_{SAVE} = T_{NSR} + \delta t_2$
$K_{NSR} = K_{NSR} + 1$
EXIT

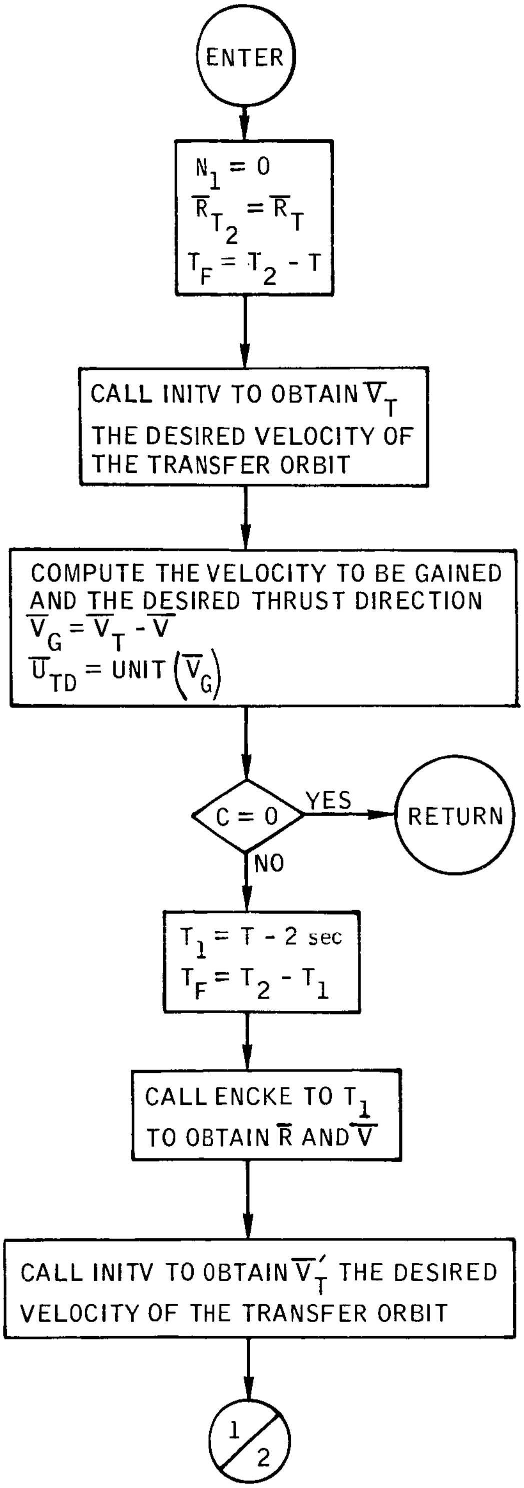
Page 2 of 2

Figure B-49.- Concluded.


ENTER
$C(\xi) = 1/2$
$S(\xi) = 1/6$
$\gamma_S = S(\xi)$
$S_M = 2$
$S_M = S_M + 2$
$\gamma_C = -\gamma_S/S_M \cdot \xi$
$\gamma_S = \gamma_C/(S_M + 1)$
$C(\xi) = C(\xi) + \gamma_C$
$S(\xi) = S(\xi) + \gamma_S$
$|\gamma_C| < 10^{-7}$
NO
YES
$S_M = 80$
NO
YES
$T_{21} = \frac{\left[C_1 x^2 C(\xi) + x \left\{C_2 x^2 \cdot S(\xi) + \left|\bar{R}(T_1)\right|\right\}\right]}{\sqrt{\mu}}$
PRINT NON-CONVERGENCE MESSAGE
RETURN

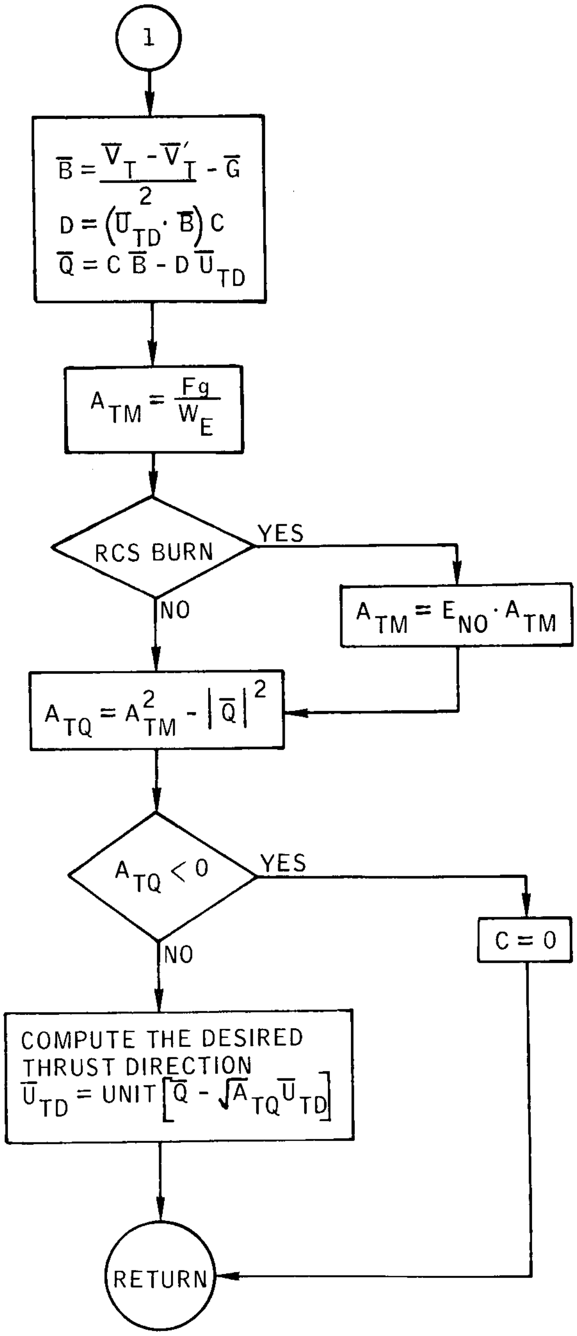
Page 1 of 1

Figure B-50.- Flow chart of subroutine KEP.


ENTER
$N_1 = 0$
$\overline{R}_{T_2} = \overline{R}_T$
$T_F = T_2 - T$
CALL INITV TO OBTAIN $\overline{V}_T$ THE DESIRED VELOCITY OF THE TRANSFER ORBIT
COMPUTE THE VELOCITY TO BE GAINED AND THE DESIRED THRUST DIRECTION
$\overline{V}_G = \overline{V}_T - \overline{V}$
$\overline{U}_{TD} = \text{UNIT}\left(\overline{V}_G\right)$
$C = 0$
YES
RETURN
NO
$T_1 = T - 2$ sec
$T_F = T_2 - T_1$
CALL ENCKE TO T_1 TO OBTAIN $\overline{R}$ AND $\overline{V}$
CALL INITV TO OBTAIN $\overline{V}_T'$ THE DESIRED VELOCITY OF THE TRANSFER ORBIT
1
2

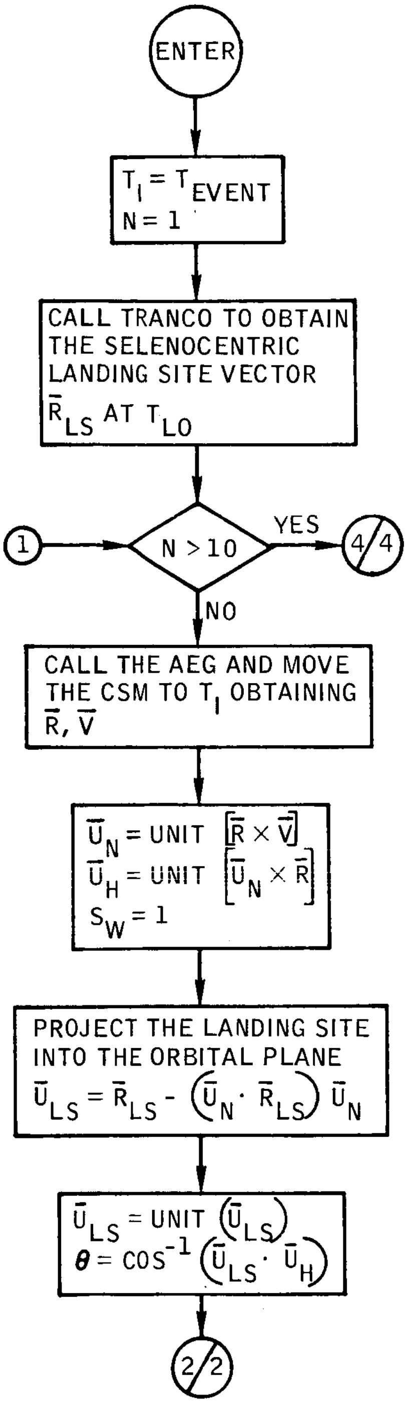
Page 1 of 2

Figure B-51.- Flow chart of subroutine LAMAIM.


1
$\overline{B} = \frac{\overline{V}_T - \overline{V}'_T}{2} - \overline{G}$
$D = \left(\overline{U}_{TD} \cdot \overline{B}\right) C$
$\overline{Q} = C\,\overline{B} - D\,\overline{U}_{TD}$
$A_{TM} = \frac{Fg}{W_E}$
RCS BURN
YES
NO
$A_{TM} = E_{NO} \cdot A_{TM}$
$A_{TQ} = A_{TM}^2 - \left|\overline{Q}\right|^2$
$A_{TQ} < 0$
YES
NO
$C = 0$
COMPUTE THE DESIRED THRUST DIRECTION
$\overline{U}_{TD} = \text{UNIT}\left[\overline{Q} - \sqrt{A_{TQ}}\,\overline{U}_{TD}\right]$
RETURN

Page 2 of 2

Figure B-51.- Concluded.


ENTER
$T_I = T_{EVENT}$
$N = 1$
CALL TRANCO TO OBTAIN THE SELENOCENTRIC LANDING SITE VECTOR $\bar{R}_{LS}$ AT T_{LO}
1
$N > 10$
YES
4/4
NO
CALL THE AEG AND MOVE THE CSM TO T_I OBTAINING $\bar{R}, \bar{V}$
$\bar{U}_N = \text{UNIT} \left[\bar{R} \times \bar{V}\right]$
$\bar{U}_H = \text{UNIT} \left[\bar{U}_N \times \bar{R}\right]$
$S_W = 1$
PROJECT THE LANDING SITE INTO THE ORBITAL PLANE
$\bar{U}_{LS} = \bar{R}_{LS} - \left(\bar{U}_N \cdot \bar{R}_{LS}\right) \bar{U}_N$
$\bar{U}_{LS} = \text{UNIT} \left(\bar{U}_{LS}\right)$
$\theta = \cos^{-1} \left(\bar{U}_{LS} \cdot \bar{U}_H\right)$
2/2

Page 1 of 4

Figure B-52.- Flow chart of subroutine LOPC.

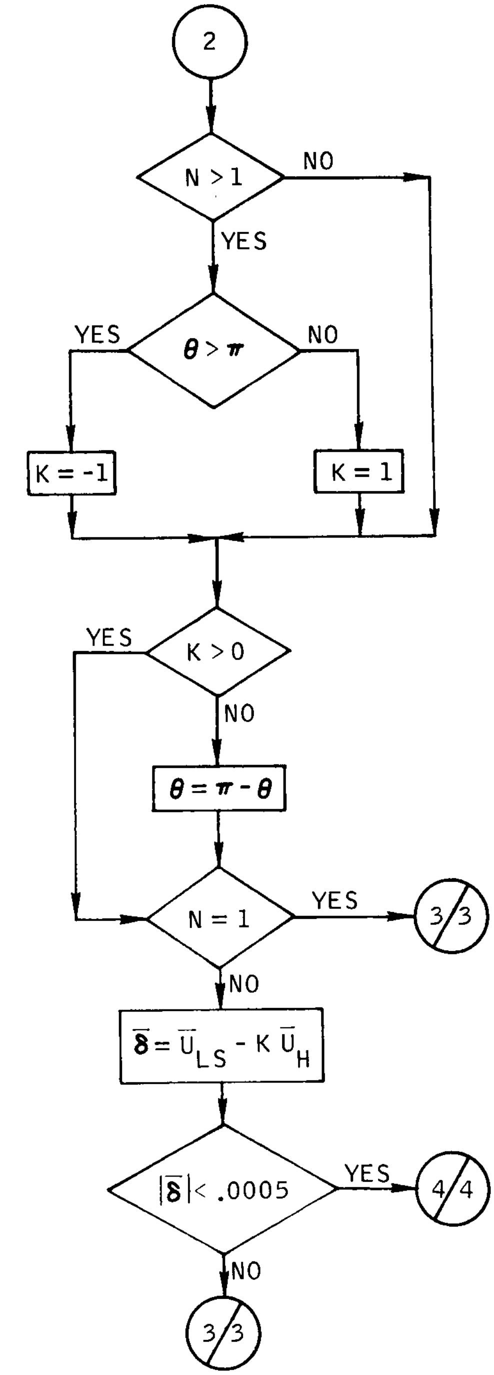
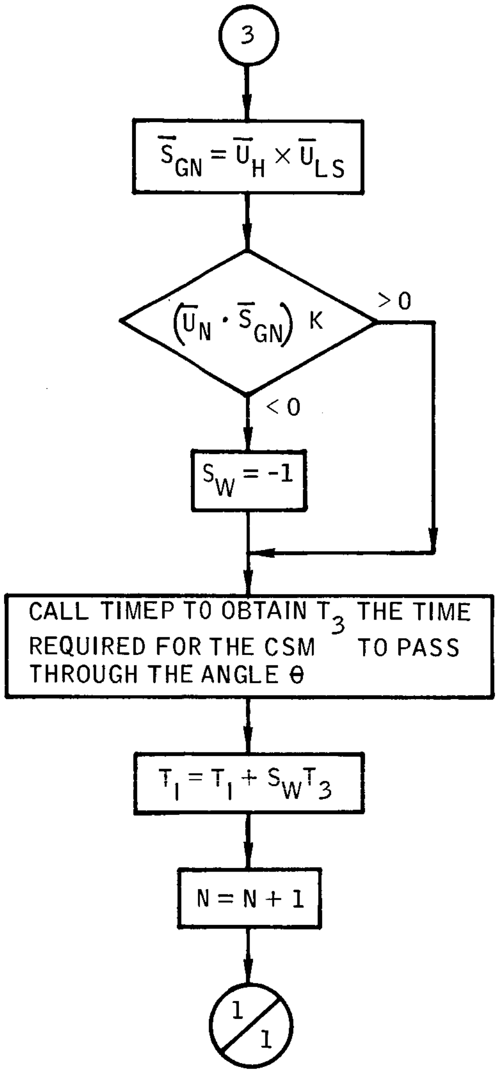


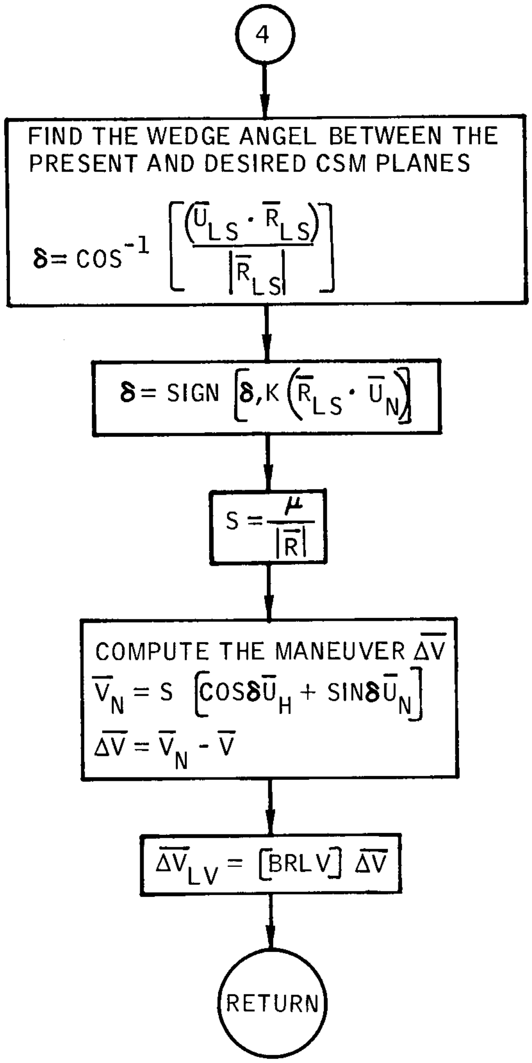
Figure B-52.- Continued.

Page 2 of 4


3
$\bar{S}_{GN} = \bar{U}_H \times \bar{U}_{LS}$
$(\bar{U}_N \cdot \bar{S}_{GN})$ K
> 0
< 0
$S_W = -1$
CALL TIMEP TO OBTAIN T_3 THE TIME REQUIRED FOR THE CSM TO PASS THROUGH THE ANGLE θ
$T_I = T_I + S_W T_3$
N = N + 1
1
1

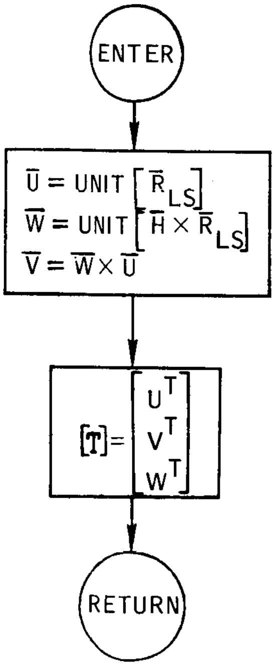
Page 3 of 4

Figure B-52.- Continued.


4
FIND THE WEDGE ANGEL BETWEEN THE PRESENT AND DESIRED CSM PLANES
$\delta = \cos^{-1}\left[\frac{(\bar{U}_{LS} \cdot \bar{R}_{LS})}{|\bar{R}_{LS}|}\right]$
$\delta = \mathrm{SIGN}\left[\delta, K(\bar{R}_{LS} \cdot \bar{U}_N)\right]$
$S = \frac{\mu}{|\bar{R}|}$
COMPUTE THE MANEUVER $\overline{\Delta V}$
$\bar{V}_N = S\ [\cos\delta \bar{U}_H + \sin\delta \bar{U}_N]$
$\overline{\Delta V} = \bar{V}_N - \bar{V}$
$\overline{\Delta V}_{LV} = [BRLV]\ \overline{\Delta V}$
RETURN

Page 4 of 4

Figure B-52.- Concluded.


ENTER
$\bar{U} = \text{UNIT}\left[\bar{R}_{LS}\right]$
$\bar{W} = \text{UNIT}\left[\bar{H} \times \bar{R}_{LS}\right]$
$\bar{V} = \bar{W} \times \bar{U}$
$[T] = \begin{bmatrix} U^T \\ V^T \\ W^T \end{bmatrix}$
RETURN

Page 1 of 1

Figure B-53.- Flow chart of subroutine LSALGN.

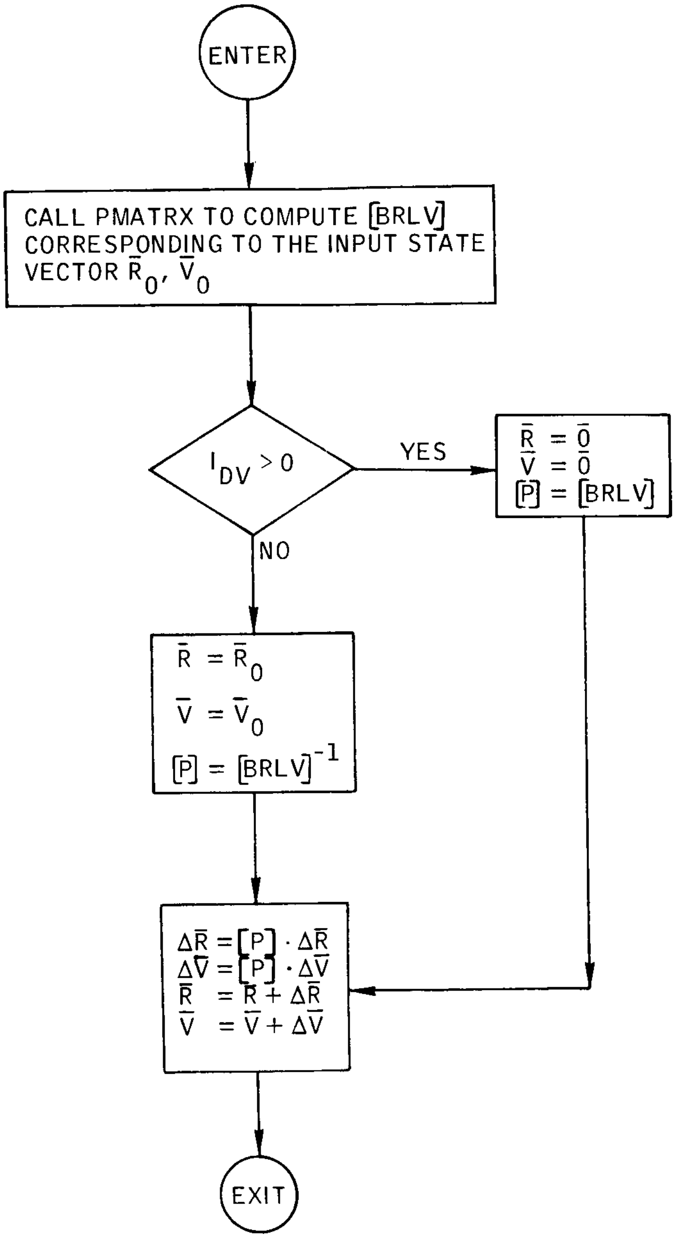

ENTER
CALL PMATRX TO COMPUTE [BRLV] CORRESPONDING TO THE INPUT STATE VECTOR $\bar{R}_0$, $\bar{V}_0$
$I_{DV} > 0$
YES
NO
$\bar{R} = \bar{0}$
$\bar{V} = \bar{0}$
$[P] = [BRLV]$
$\bar{R} = \bar{R}_0$
$\bar{V} = \bar{V}_0$
$[P] = [BRLV]^{-1}$
$\Delta\bar{R} = [P] \cdot \Delta\bar{R}$
$\Delta\bar{V} = [P] \cdot \Delta\bar{V}$
$\bar{R} = \bar{R} + \Delta\bar{R}$
$\bar{V} = \bar{V} + \Delta\bar{V}$
EXIT

Figure B-54.- Flow chart of subroutine LVLHDX. Page 1 of 1

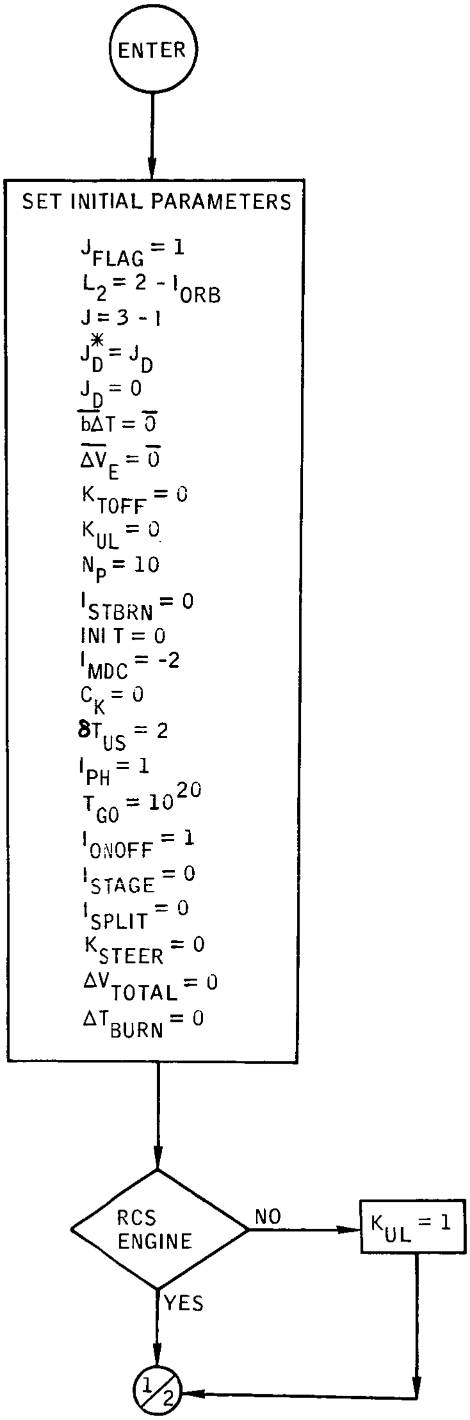
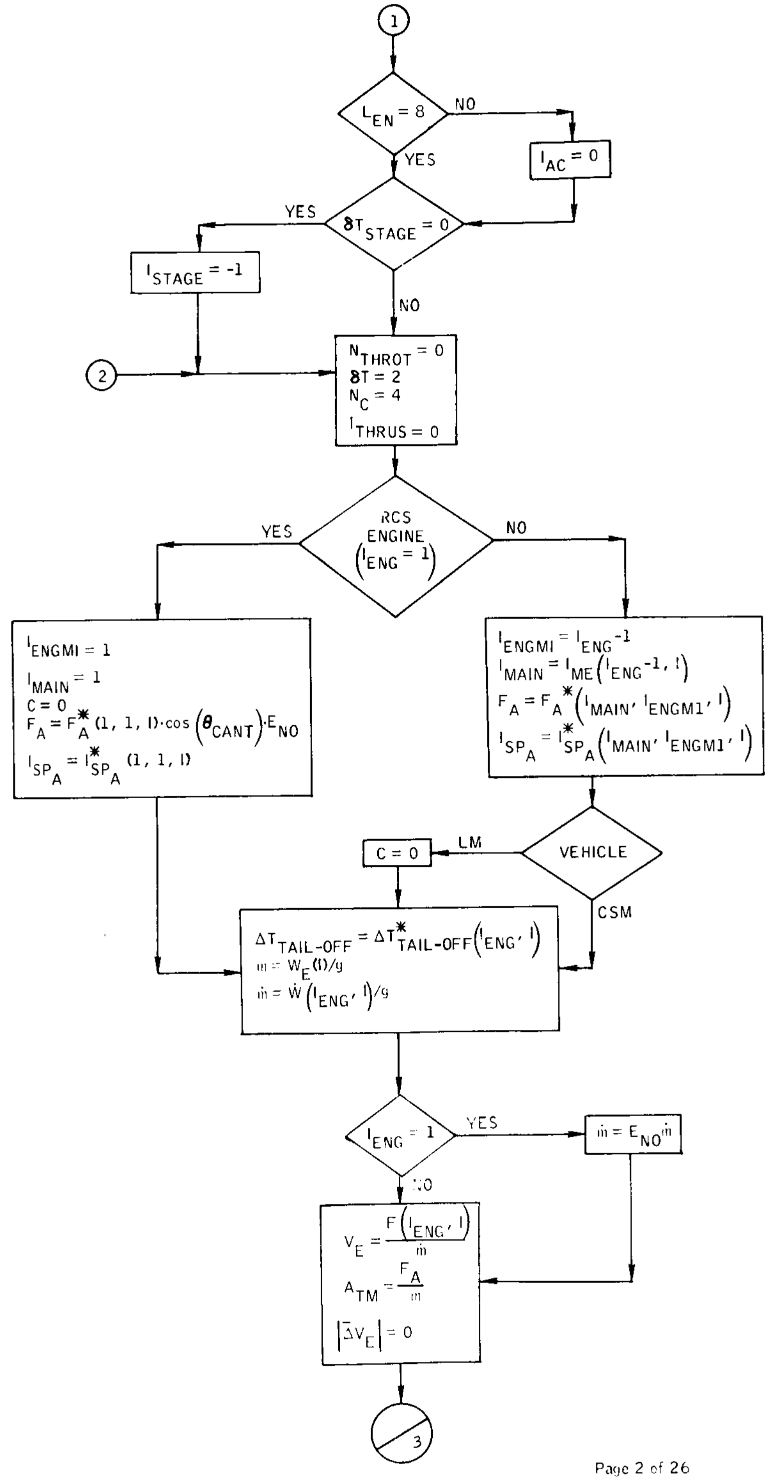


Figure B-55.- Flow chart of subroutine MANEUV. Page 1 of 26


1
$L_{EN} = 8$
NO
YES
$I_{AC} = 0$
YES
$\delta T_{STAGE} = 0$
$I_{STAGE} = -1$
NO
2
$N_{THROT} = 0$
$\delta T = 2$
$N_C = 4$
$I_{THRUS} = 0$
YES
RCS ENGINE $(I_{ENG} = 1)$
NO
$I_{ENGMI} = 1$
$I_{MAIN} = 1$
$C = 0$
$F_A = F_A^*(1, 1, I) \cdot \cos(\theta_{CANT}) \cdot E_{NO}$
$I_{SP_A} = I_{SP_A}^*(1, 1, I)$
$I_{ENGMI} = I_{ENG} - 1$
$I_{MAIN} = I_{ME}(I_{ENG} - 1, I)$
$F_A = F_A^*(I_{MAIN}, I_{ENGM1}, I)$
$I_{SP_A} = I_{SP_A}^*(I_{MAIN}, I_{ENGM1}, I)$
$C = 0$
LM
VEHICLE
CSM
$\Delta T_{TAIL\text{-}OFF} = \Delta T_{TAIL\text{-}OFF}^*(I_{ENG}, I)$
$m = W_E(I)/g$
$\dot{m} = \dot{W}(I_{ENG}, I)/g$
$I_{ENG} = 1$
YES
$\dot{m} = E_{NO}\dot{m}$
NO
$V_E = \frac{F(I_{ENG}, I)}{\dot{m}}$
$A_{TM} = \frac{F_A}{m}$
$|\bar{\Delta V}_E| = 0$
3

Page 2 of 26

Figure B-55.- Continued.

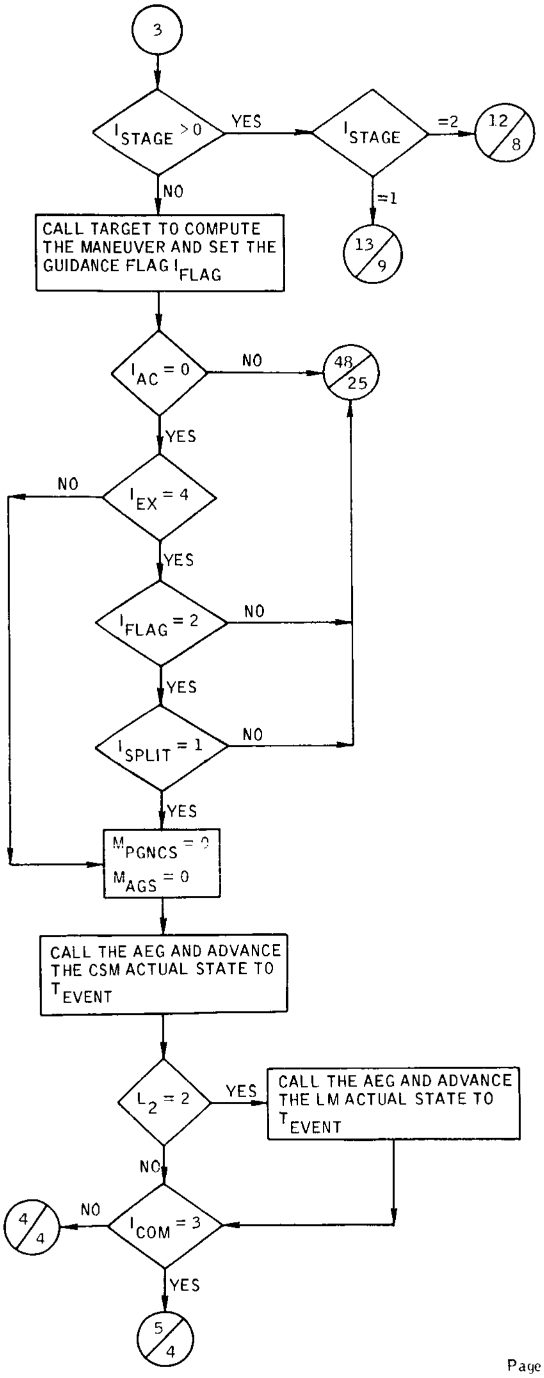
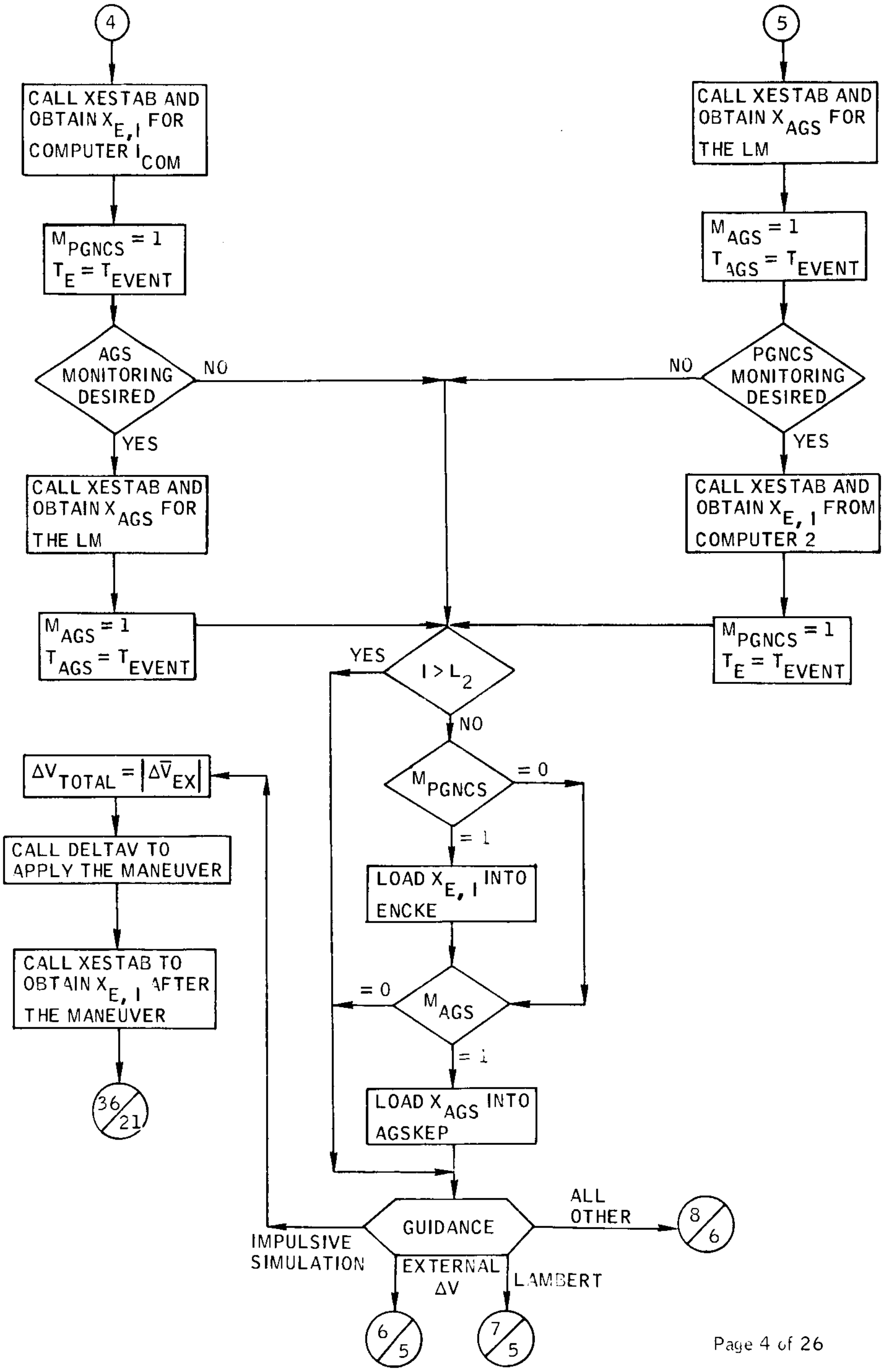
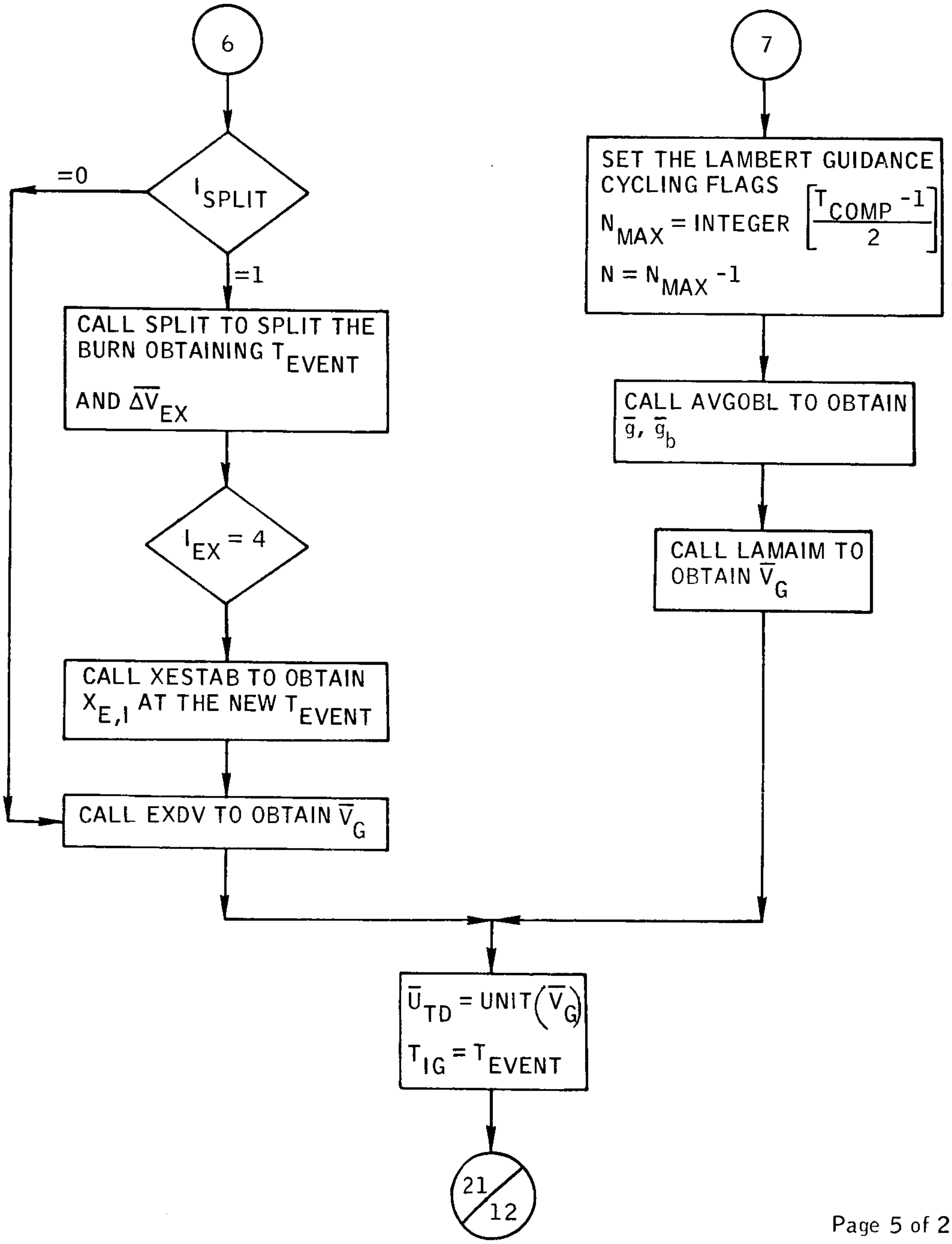

3
$I_{STAGE} > 0$
YES
I_{STAGE}
=2
12/8
=1
13/9
NO
CALL TARGET TO COMPUTE THE MANEUVER AND SET THE GUIDANCE FLAG I_{FLAG}
$I_{AC} = 0$
NO
48/25
YES
$I_{EX} = 4$
NO
YES
$I_{FLAG} = 2$
NO
YES
$I_{SPLIT} = 1$
NO
YES
$M_{PGNCS} = 0$
$M_{AGS} = 0$
CALL THE AEG AND ADVANCE THE CSM ACTUAL STATE TO T_{EVENT}
$L_2 = 2$
YES
CALL THE AEG AND ADVANCE THE LM ACTUAL STATE TO T_{EVENT}
NO
$I_{COM} = 3$
NO
4/4
YES
5/4

Figure B-55.- Continued.



Page 4 of 26

Figure B-55.- Continued.


6
7
I_{SPLIT}
=0
=1
CALL SPLIT TO SPLIT THE BURN OBTAINING T_{EVENT} AND $\overline{\Delta V}_{EX}$
$I_{EX} = 4$
CALL XESTAB TO OBTAIN $X_{E,I}$ AT THE NEW T_{EVENT}
CALL EXDV TO OBTAIN $\overline{V}_G$
SET THE LAMBERT GUIDANCE CYCLING FLAGS
$N_{MAX} = \text{INTEGER}\left[\frac{T_{COMP} - 1}{2}\right]$
$N = N_{MAX} - 1$
CALL AVGOBL TO OBTAIN $\overline{g}$, $\overline{g}_b$
CALL LAMAIM TO OBTAIN $\overline{V}_G$
$\overline{U}_{TD} = \text{UNIT}(\overline{V}_G)$
$T_{IG} = T_{EVENT}$
21/12

Page 5 of 26

Figure B-55.- Continued.

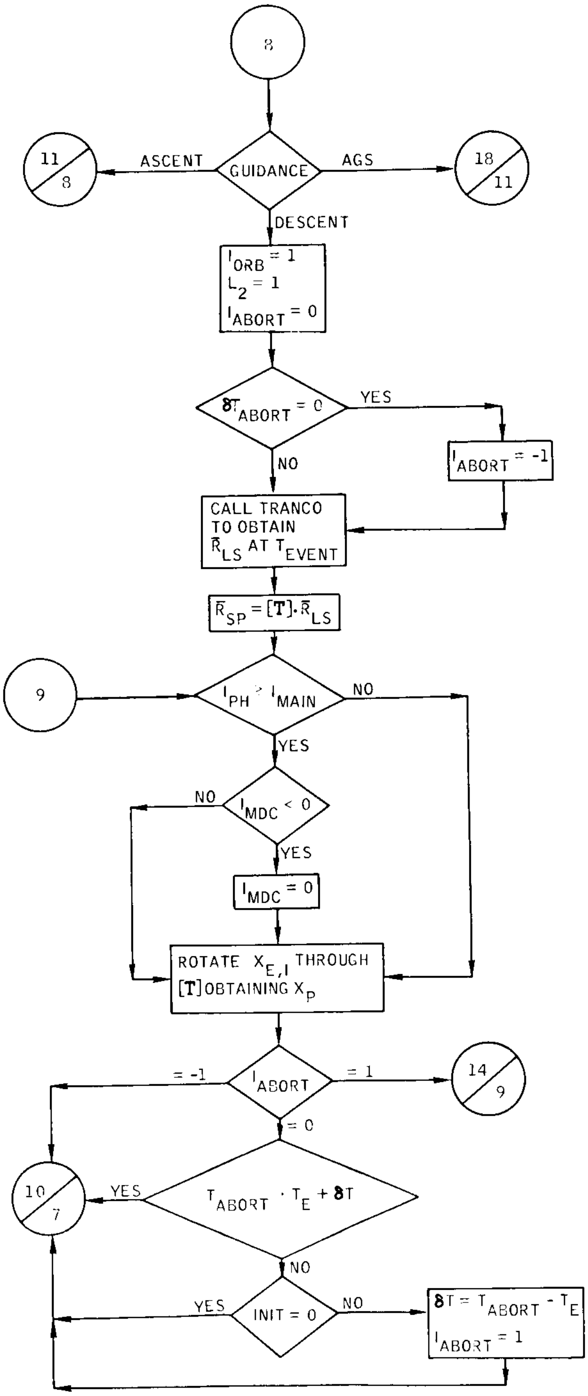


Figure B 55.- Continued.

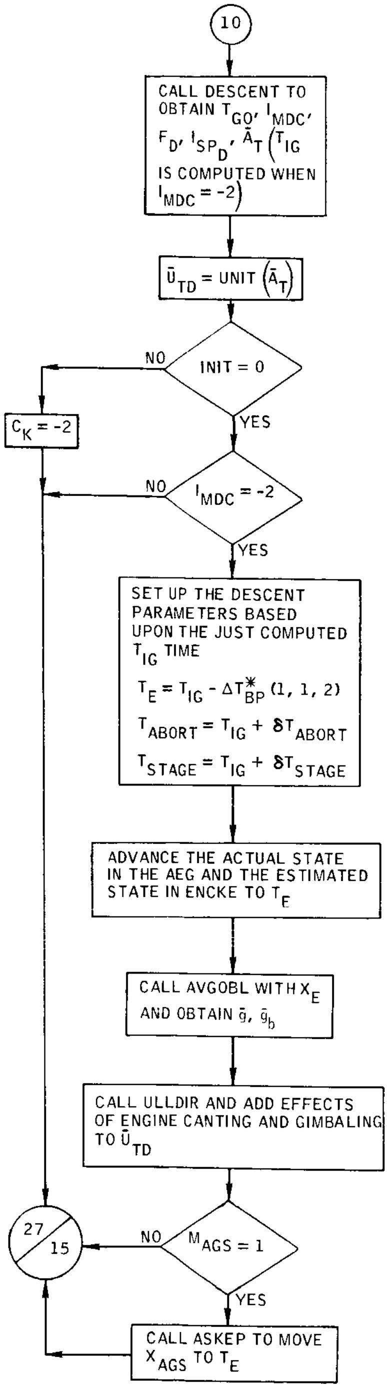

10
CALL DESCENT TO OBTAIN T_{GO}, I_{MDC}, F_D, I_{SP_D}, $\bar{A}_T$ (T_{IG} IS COMPUTED WHEN $I_{MDC} = -2$)
$\bar{U}_{TD}$ = UNIT $(\bar{A}_T)$
INIT = 0
NO
YES
$C_K = -2$
$I_{MDC} = -2$
NO
YES
SET UP THE DESCENT PARAMETERS BASED UPON THE JUST COMPUTED T_{IG} TIME
$T_E = T_{IG} - \Delta T^*_{BP}(1, 1, 2)$
$T_{ABORT} = T_{IG} + \delta T_{ABORT}$
$T_{STAGE} = T_{IG} + \delta T_{STAGE}$
ADVANCE THE ACTUAL STATE IN THE AEG AND THE ESTIMATED STATE IN ENCKE TO T_E
CALL AVGOBL WITH X_E AND OBTAIN $\bar{g}$, $\bar{g}_b$
CALL ULLDIR AND ADD EFFECTS OF ENGINE CANTING AND GIMBALING TO $\bar{U}_{TD}$
$M_{AGS} = 1$
NO
YES
27
15
CALL ASKEP TO MOVE X_{AGS} TO T_E

Figure B-55.- Continued.

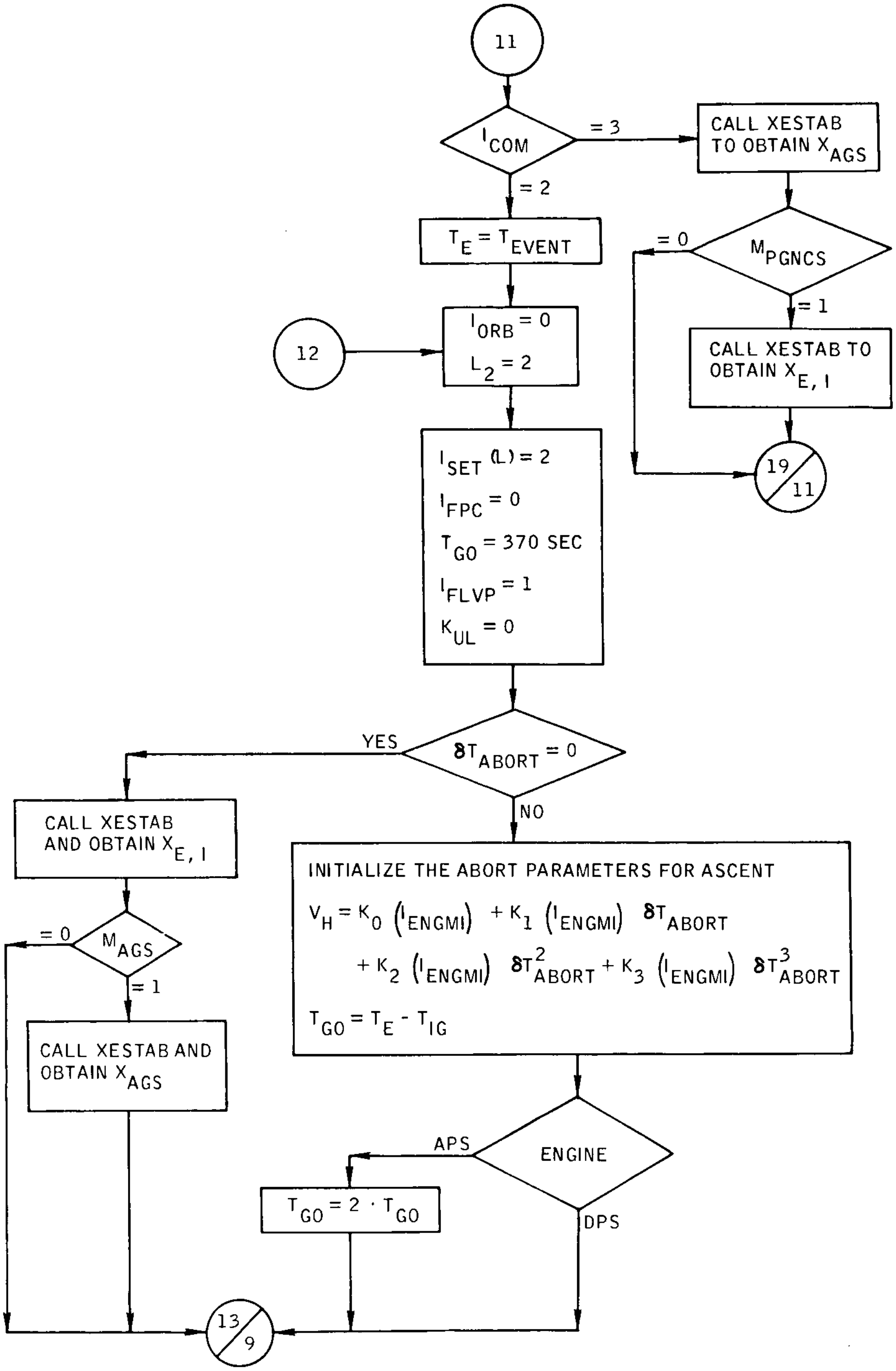
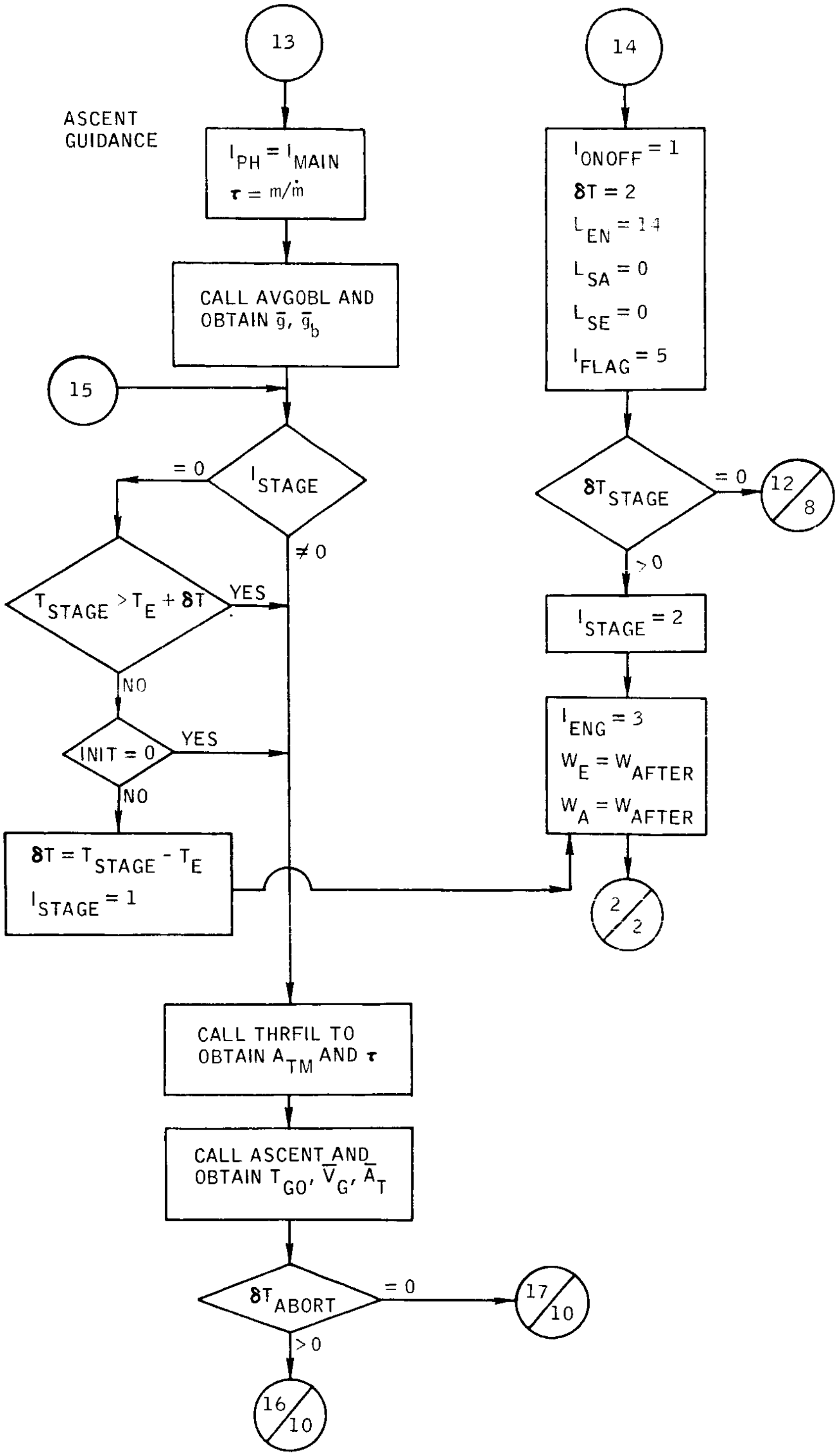

11
I_{COM}
= 3
CALL XESTAB TO OBTAIN X_{AGS}
= 2
$T_E = T_{EVENT}$
M_{PGNCS}
= 0
= 1
CALL XESTAB TO OBTAIN $X_{E,I}$
12
$I_{ORB} = 0$
$L_2 = 2$
19/11
$I_{SET}(L) = 2$
$I_{FPC} = 0$
$T_{GO} = 370$ SEC
$I_{FLVP} = 1$
$K_{UL} = 0$
YES
$\delta T_{ABORT} = 0$
NO
CALL XESTAB AND OBTAIN $X_{E,I}$
INITIALIZE THE ABORT PARAMETERS FOR ASCENT
$V_H = K_0 (I_{ENGMI}) + K_1 (I_{ENGMI}) \delta T_{ABORT} + K_2 (I_{ENGMI}) \delta T^2_{ABORT} + K_3 (I_{ENGMI}) \delta T^3_{ABORT}$
$T_{GO} = T_E - T_{IG}$
M_{AGS}
= 0
= 1
CALL XESTAB AND OBTAIN X_{AGS}
APS
ENGINE
DPS
$T_{GO} = 2 \cdot T_{GO}$
13/9

Figure B-55.- Continued.

Page 8 of 26



Page 9 of 26

Figure B-55.- Continued.

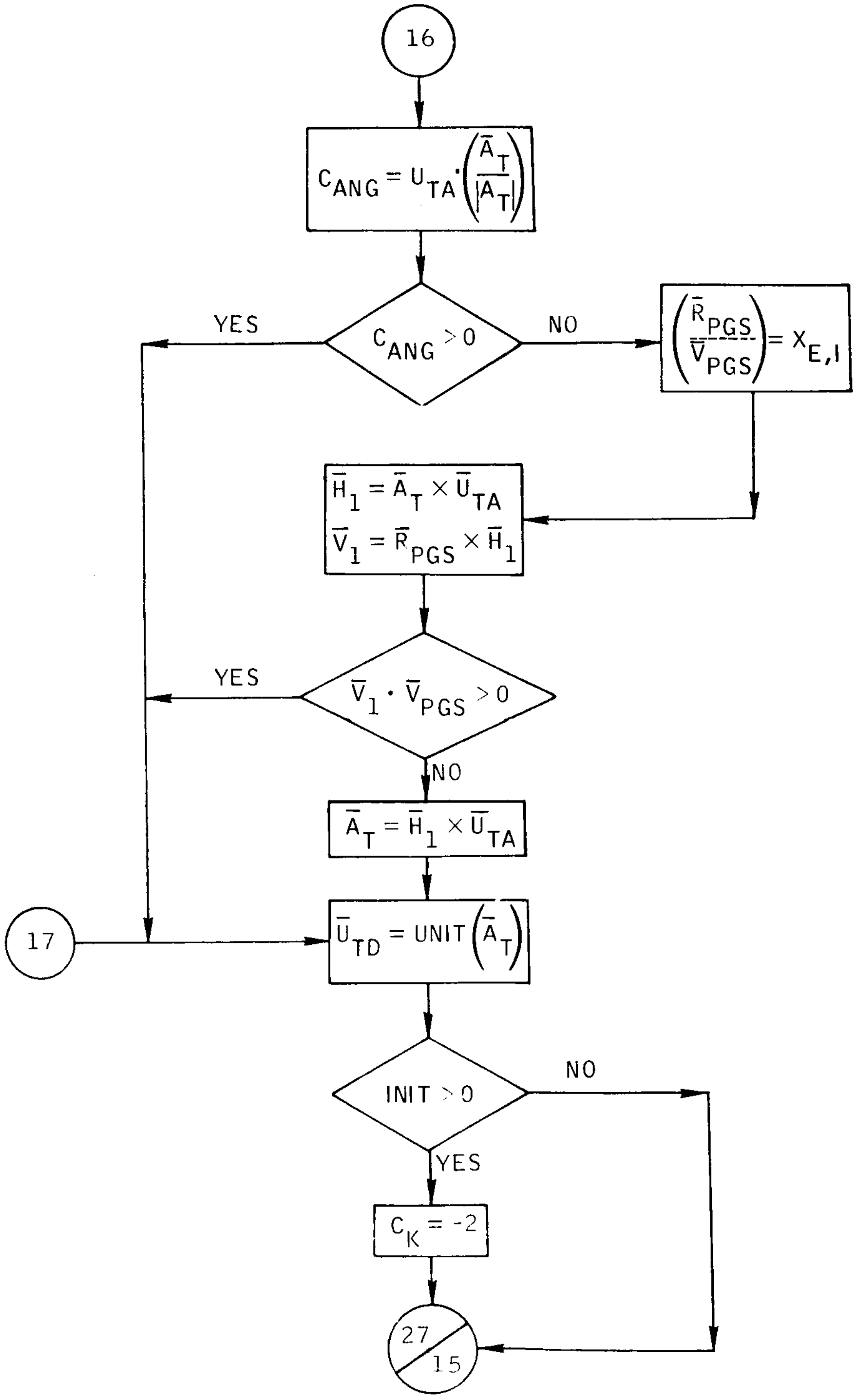
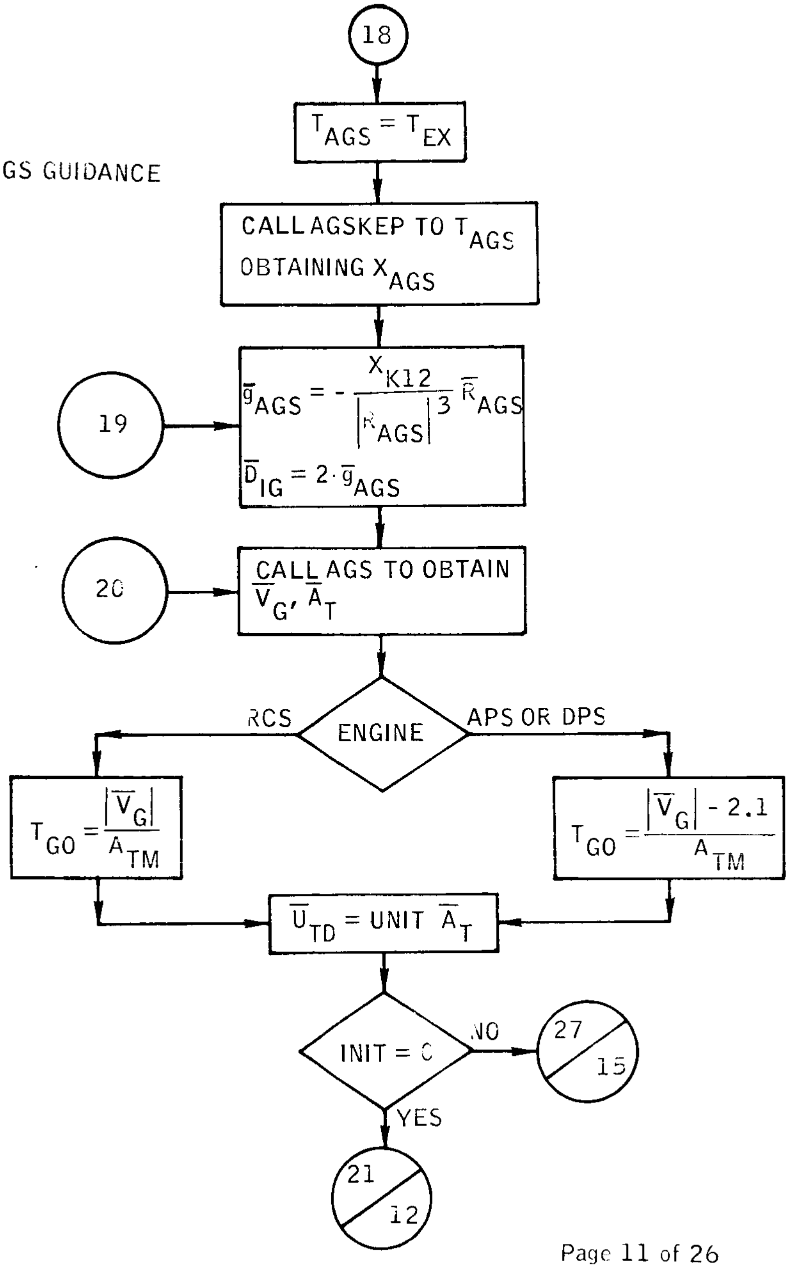

16
$C_{ANG} = U_{TA} \cdot \left(\frac{\bar{A}_T}{|A_T|}\right)$
$C_{ANG} > 0$
YES
NO
$\left(\frac{\bar{R}_{PGS}}{\bar{V}_{PGS}}\right) = X_{E,I}$
$\bar{H}_1 = \bar{A}_T \times \bar{U}_{TA}$
$\bar{V}_1 = \bar{R}_{PGS} \times \bar{H}_1$
$\bar{V}_1 \cdot \bar{V}_{PGS} > 0$
YES
NO
$\bar{A}_T = \bar{H}_1 \times \bar{U}_{TA}$
17
$\bar{U}_{TD} = \text{UNIT}\left(\bar{A}_T\right)$
$\text{INIT} > 0$
NO
YES
$C_K = -2$
27
15

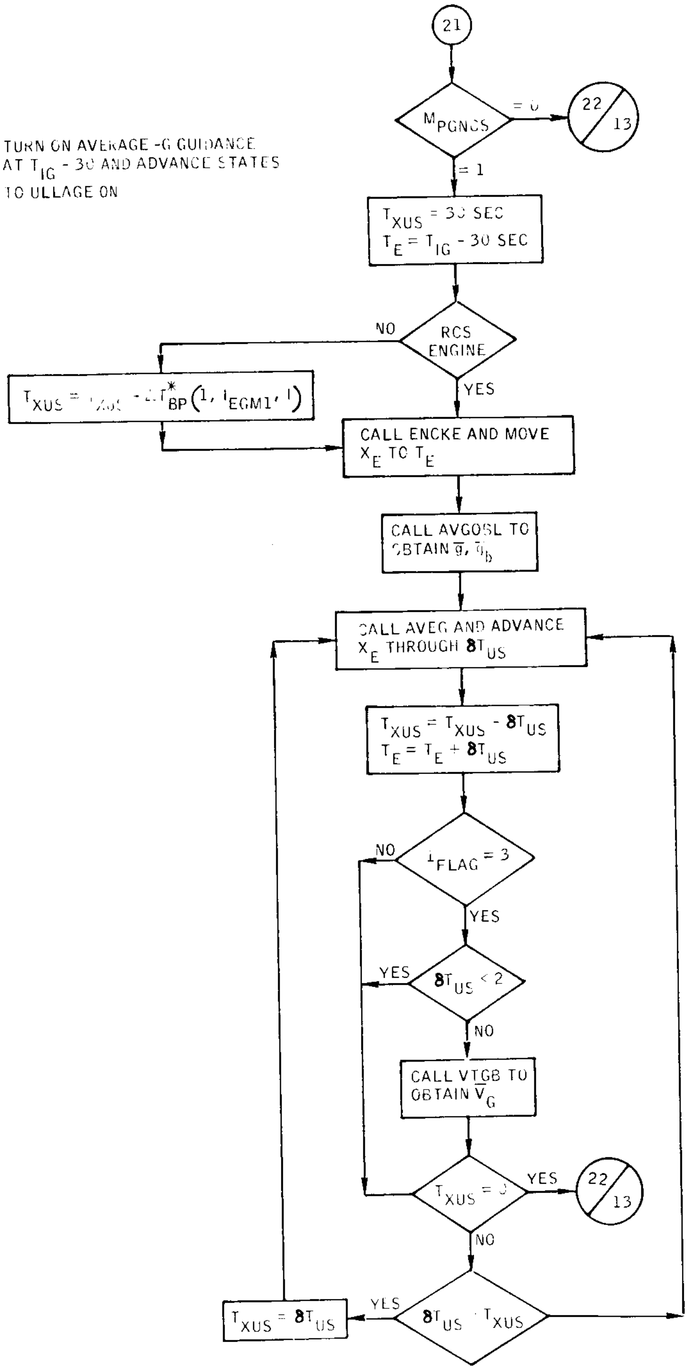
Figure B-55.- Continued.

AGS GUIDANCE


18
$T_{AGS} = T_{EX}$
CALL AGSKEP TO T_{AGS} OBTAINING X_{AGS}
19
$\bar{g}_{AGS} = -\frac{X_{K12}}{|\bar{R}_{AGS}|^3}\bar{R}_{AGS}$
$\bar{D}_{IG} = 2 \cdot \bar{g}_{AGS}$
20
CALL AGS TO OBTAIN $\bar{V}_G, \bar{A}_T$
ENGINE
RCS
APS OR DPS
$T_{GO} = \frac{|\bar{V}_G|}{A_{TM}}$
$T_{GO} = \frac{|\bar{V}_G| - 2.1}{A_{TM}}$
$\bar{U}_{TD}$ = UNIT $\bar{A}_T$
INIT = 0
NO
27/15
YES
21/12

Page 11 of 26

Figure B-55.- Continued.


TURN ON AVERAGE -G GUIDANCE AT T_{IG} - 30 AND ADVANCE STATES TO ULLAGE ON

21
M_{PGNCS}
= 0
22/13
= 1
T_{XUS} = 30 SEC
$T_E = T_{IG}$ - 30 SEC
RCS ENGINE
NO
YES
$T_{XUS} = T_{XUS} - \Delta T^*_{BP}(1, I_{EGM1}, I)$
CALL ENCKE AND MOVE X_E TO T_E
CALL AVGOSL TO OBTAIN $\bar{g}$, $\bar{g}_b$
CALL AVEG AND ADVANCE X_E THROUGH δT_{US}
$T_{XUS} = T_{XUS} - \delta T_{US}$
$T_E = T_E + \delta T_{US}$
$I_{FLAG} = 3$
NO
YES
$\delta T_{US} < 2$
YES
NO
CALL VTGB TO OBTAIN $\bar{V}_G$
$T_{XUS} = 0$
YES
22/13
NO
$\delta T_{US} > T_{XUS}$
YES
$T_{XUS} = \delta T_{US}$

Figure B 55.- Continued.

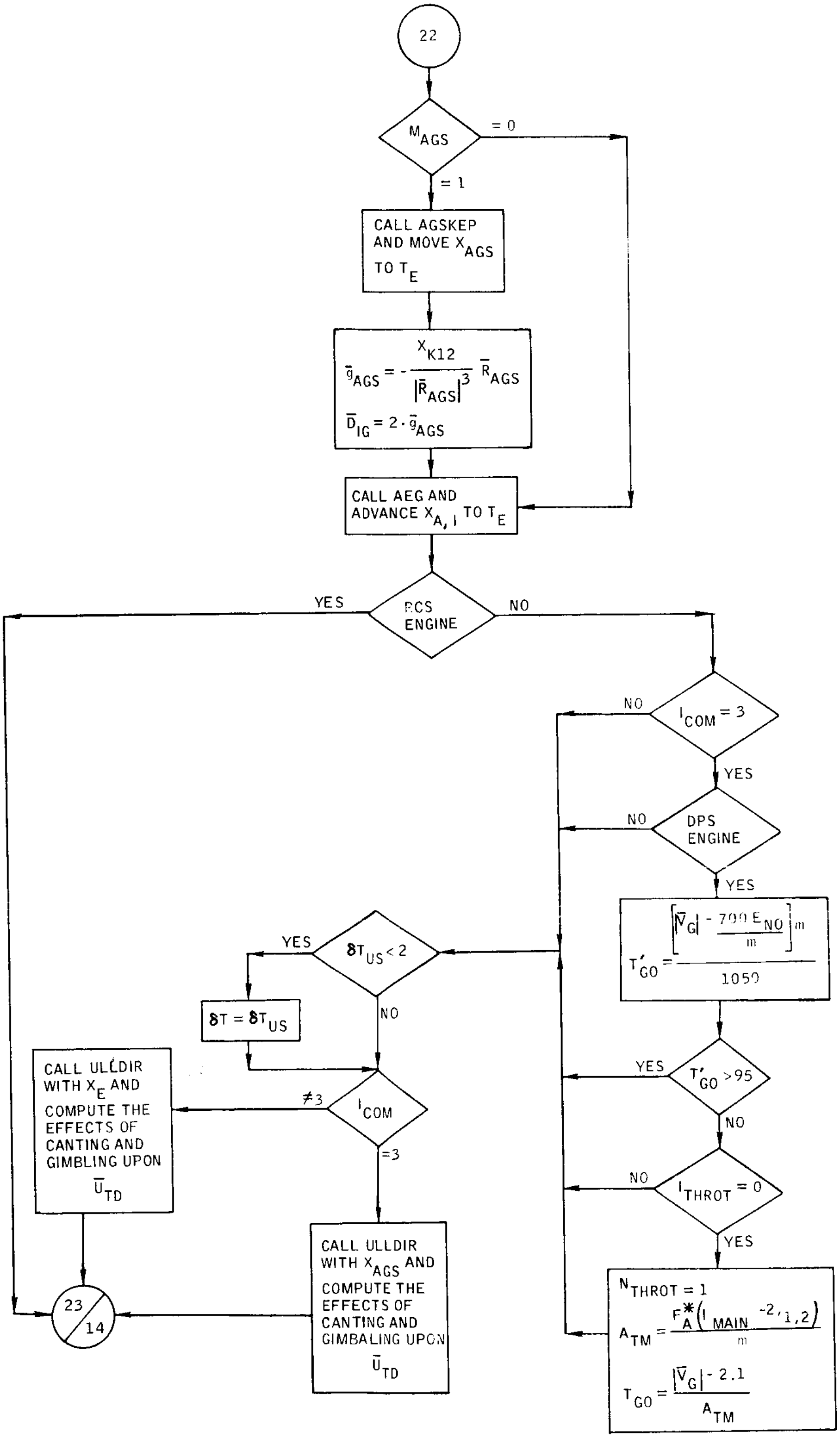

22
M_{AGS}
= 0
= 1
CALL AGSKEP AND MOVE X_{AGS} TO T_E
$\bar{g}_{AGS} = -\frac{X_{K12}}{|\bar{R}_{AGS}|^3} \bar{R}_{AGS}$
$\bar{D}_{IG} = 2 \cdot \bar{g}_{AGS}$
CALL AEG AND ADVANCE $X_{A,1}$ TO T_E
YES
RCS ENGINE
NO
NO
$I_{COM} = 3$
YES
NO
DPS ENGINE
YES
$T'_{GO} = \frac{\left[|\bar{V}_G| - \frac{700 E_{NO}}{m}\right] m}{1050}$
YES
$T'_{GO} > 95$
NO
NO
$I_{THROT} = 0$
YES
$N_{THROT} = 1$
$A_{TM} = \frac{F^*_A(I_{MAIN} - 2, 1, 2)}{m}$
$T_{GO} = \frac{|\bar{V}_G| - 2.1}{A_{TM}}$
YES
$\delta T_{US} < 2$
NO
$\delta T = \delta T_{US}$
$\neq 3$
I_{COM}
=3
CALL ULLDIR WITH X_E AND COMPUTE THE EFFECTS OF CANTING AND GIMBLING UPON $\bar{U}_{TD}$
CALL ULLDIR WITH X_{AGS} AND COMPUTE THE EFFECTS OF CANTING AND GIMBALING UPON $\bar{U}_{TD}$
23
14

Figure B-55.- Continued.

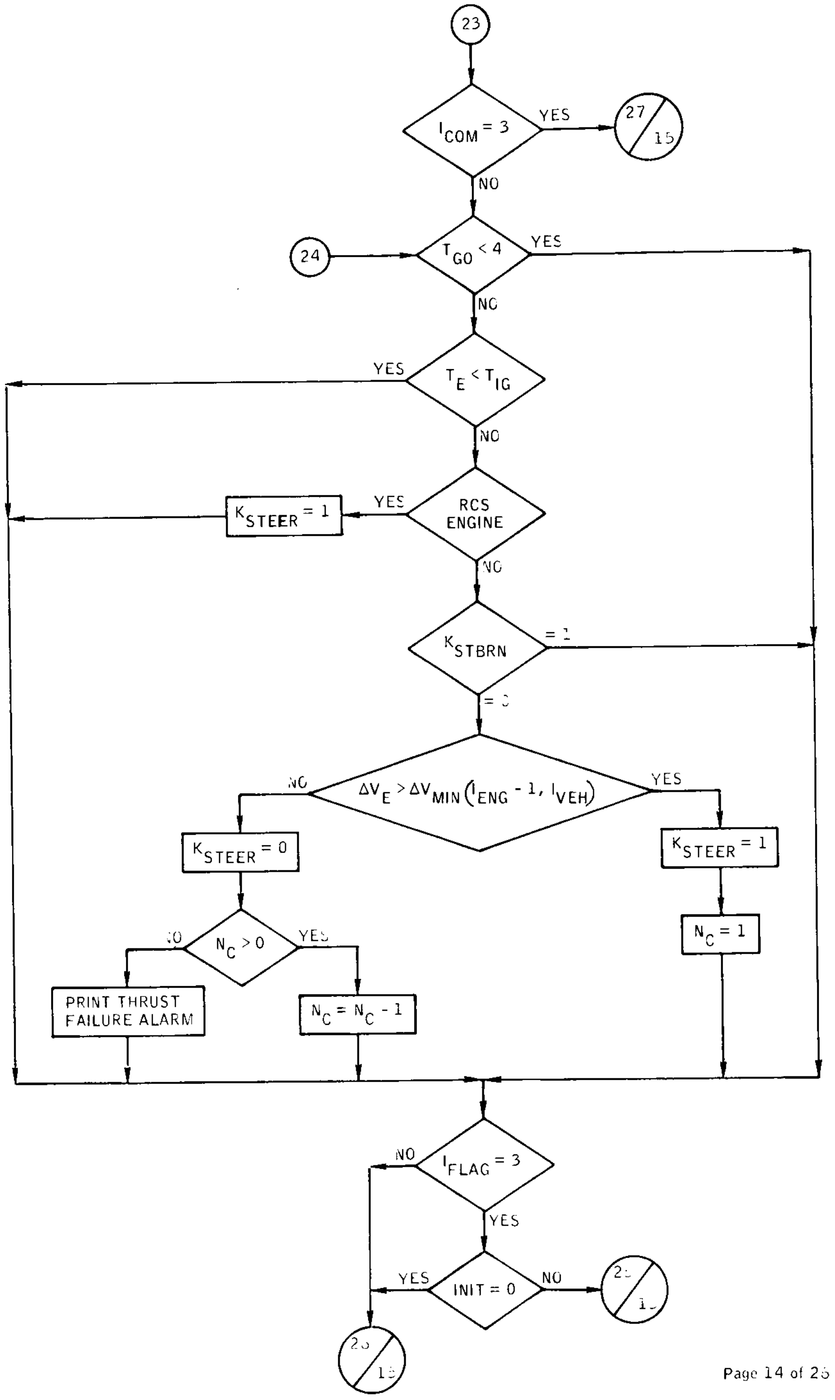


Figure B-55.- Continued.

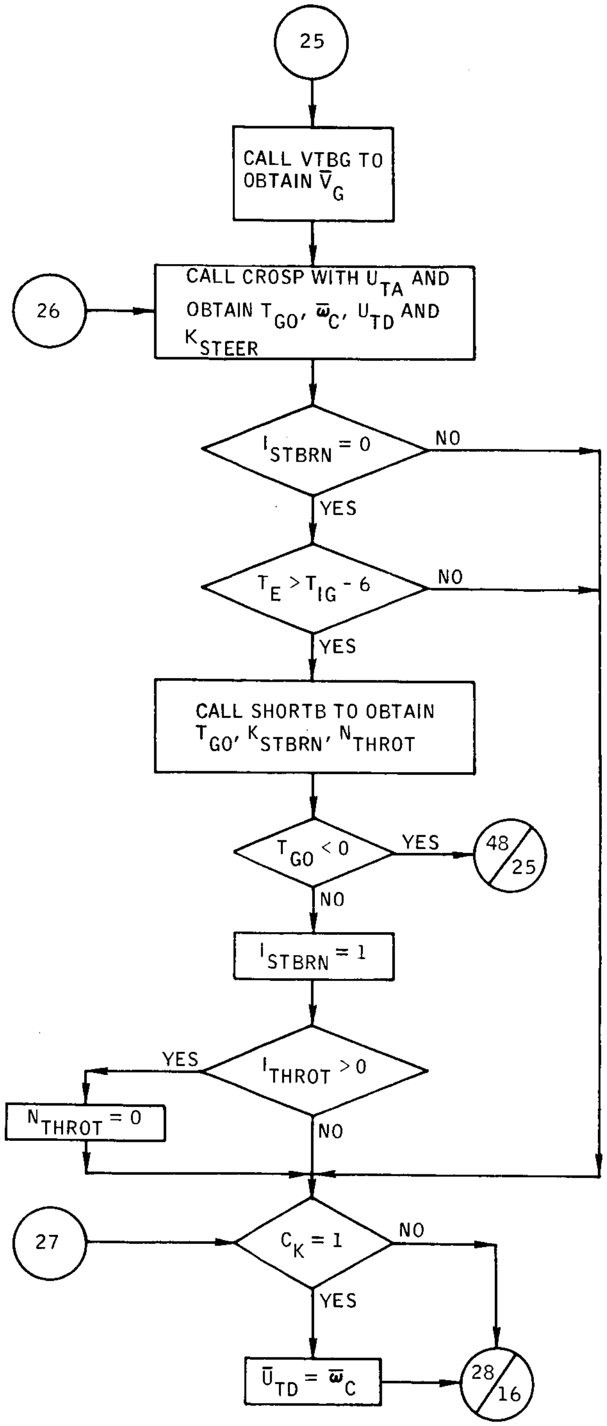
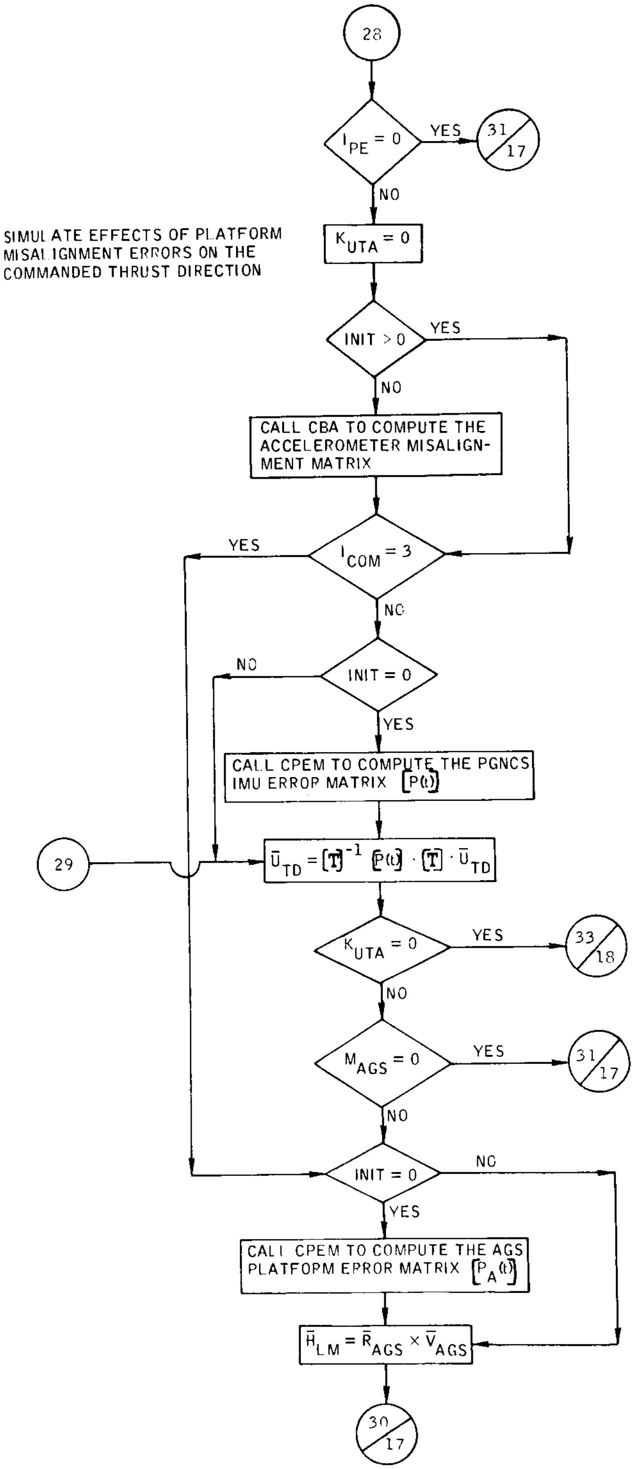

25
CALL VTBG TO OBTAIN $\bar{V}_G$
26
CALL CROSP WITH U_{TA} AND OBTAIN T_{GO}, $\bar{\omega}_C$, U_{TD} AND K_{STEER}
$I_{STBRN} = 0$
NO
YES
$T_E > T_{IG} - 6$
NO
YES
CALL SHORTB TO OBTAIN T_{GO}, K_{STBRN}, N_{THROT}
$T_{GO} < 0$
YES
48/25
NO
$I_{STBRN} = 1$
YES
$I_{THROT} > 0$
$N_{THROT} = 0$
NO
27
$C_K = 1$
NO
YES
$\bar{U}_{TD} = \bar{\omega}_C$
28/16

Figure B-55.- Continued.


28
$I_{PE} = 0$
YES
31/17
NO
$K_{UTA} = 0$
INIT > 0
YES
NO
CALL CBA TO COMPUTE THE ACCELEROMETER MISALIGNMENT MATRIX
$I_{COM} = 3$
YES
NO
INIT = 0
NO
YES
CALL CPEM TO COMPUTE THE PGNCS IMU ERROP MATRIX $[P(t)]$
29
$\bar{U}_{TD} = [T]^{-1} [P(t)] \cdot [T] \cdot \bar{U}_{TD}$
$K_{UTA} = 0$
YES
33/18
NO
$M_{AGS} = 0$
YES
31/17
NO
INIT = 0
NO
YES
CALL CPEM TO COMPUTE THE AGS PLATFOPM EPROR MATRIX $[P_A(t)]$
$\bar{H}_{LM} = \bar{R}_{AGS} \times \bar{V}_{AGS}$
30/17

SIMULATE EFFECTS OF PLATFORM MISALIGNMENT ERRORS ON THE COMMANDED THRUST DIRECTION

Page 16 of 26

Figure B-55.- Continued.

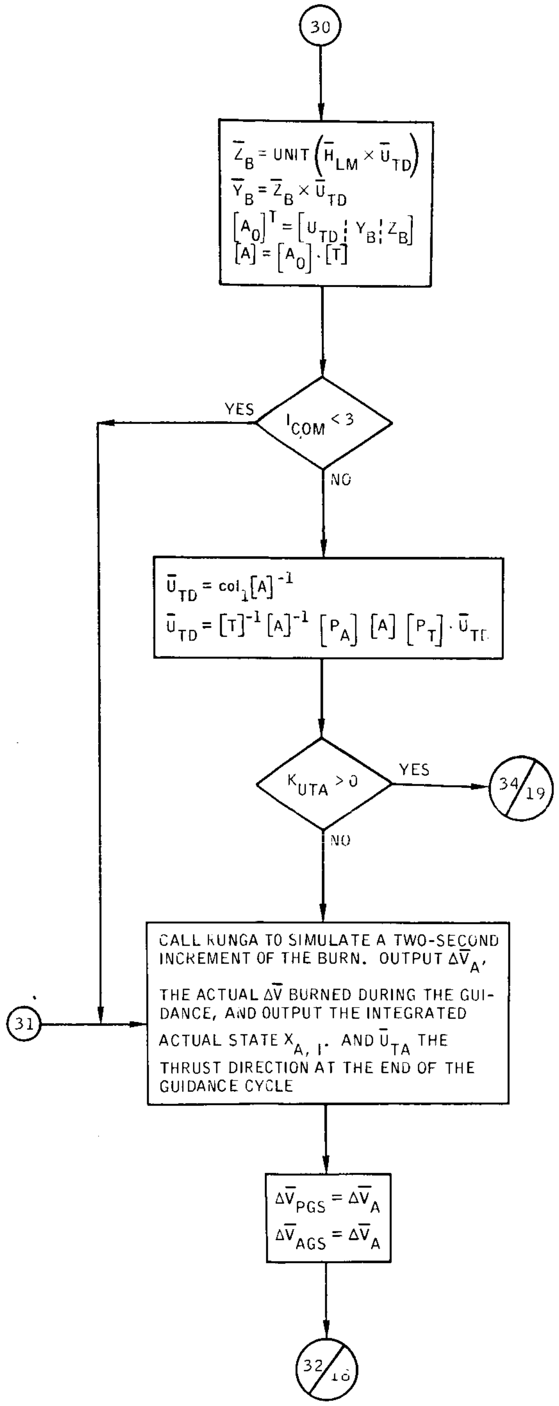
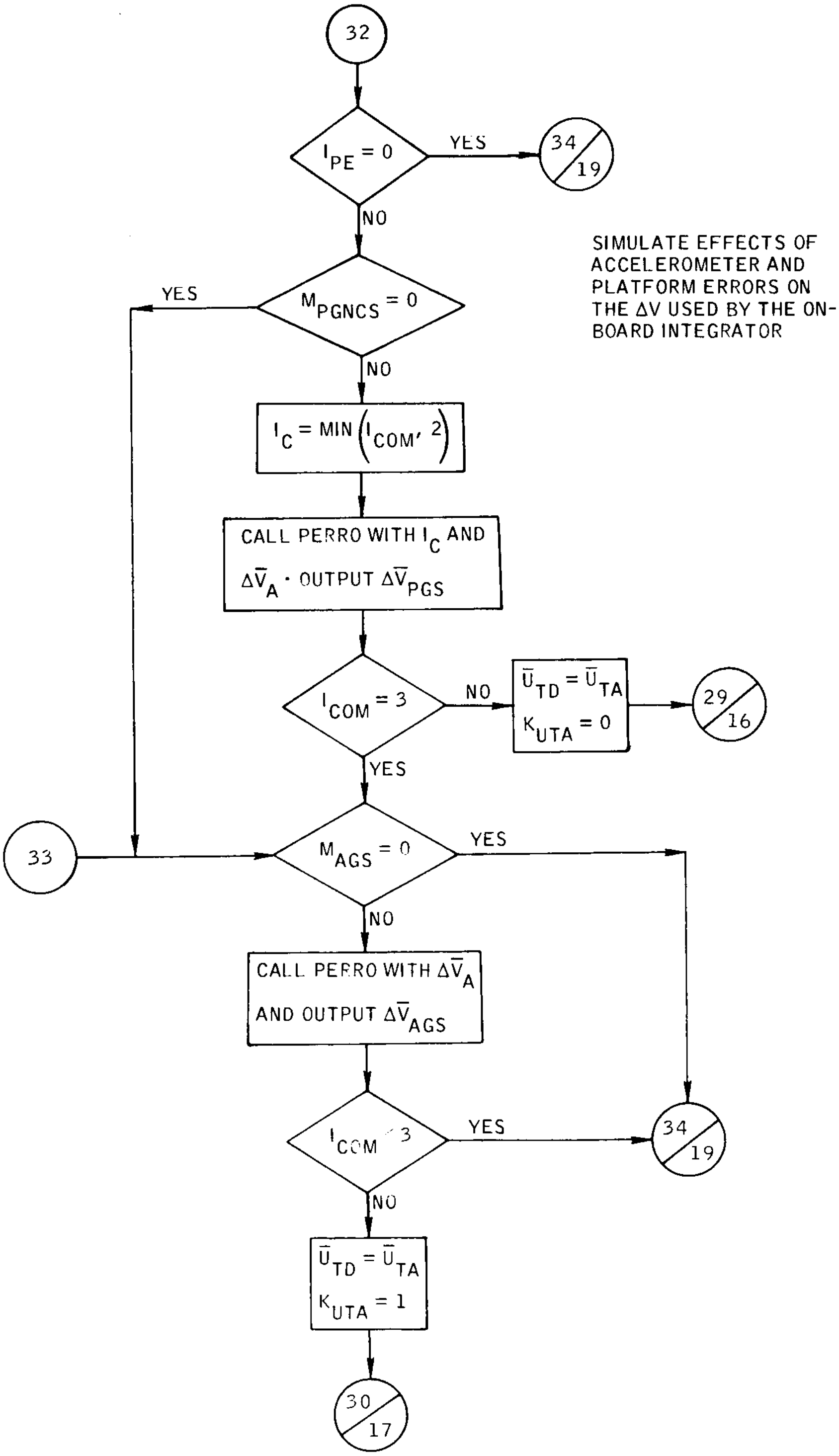

30
$\bar{Z}_B = \text{UNIT}\left(\bar{H}_{LM} \times \bar{U}_{TD}\right)$
$\bar{Y}_B = \bar{Z}_B \times \bar{U}_{TD}$
$\left[A_0\right]^T = \left[U_{TD} \vdots Y_B \vdots Z_B\right]$
$\left[A\right] = \left[A_0\right] \cdot \left[T\right]$
YES
$I_{COM} < 3$
NO
$\bar{U}_{TD} = \text{col}_1 \left[A\right]^{-1}$
$\bar{U}_{TD} = \left[T\right]^{-1} \left[A\right]^{-1} \left[P_A\right] \left[A\right] \left[P_T\right] \cdot \bar{U}_{TD}$
$K_{UTA} > 0$
YES
34/19
NO
31
CALL RUNGA TO SIMULATE A TWO-SECOND INCREMENT OF THE BURN. OUTPUT $\Delta\bar{V}_A$, THE ACTUAL $\Delta\bar{V}$ BURNED DURING THE GUIDANCE, AND OUTPUT THE INTEGRATED ACTUAL STATE $X_{A,I}$. AND $\bar{U}_{TA}$ THE THRUST DIRECTION AT THE END OF THE GUIDANCE CYCLE
$\Delta\bar{V}_{PGS} = \Delta\bar{V}_A$
$\Delta\bar{V}_{AGS} = \Delta\bar{V}_A$
32/18

Figure B-55.- Continued.



SIMULATE EFFECTS OF ACCELEROMETER AND PLATFORM ERRORS ON THE ΔV USED BY THE ON-BOARD INTEGRATOR

Figure B-55.- Continued.

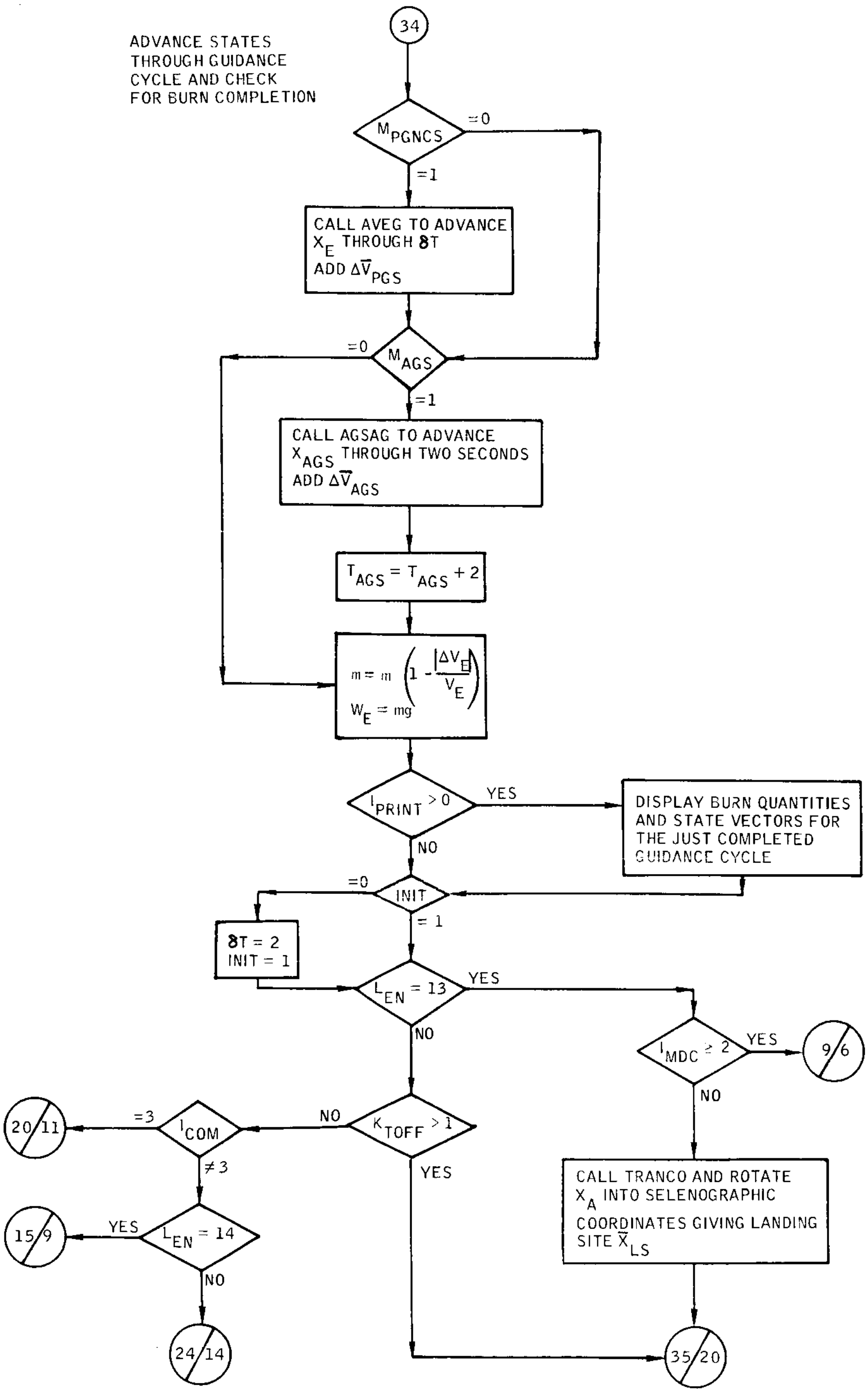

ADVANCE STATES THROUGH GUIDANCE CYCLE AND CHECK FOR BURN COMPLETION
34
M_{PGNCS}
=0
=1
CALL AVEG TO ADVANCE X_E THROUGH δT ADD $\Delta\overline{V}_{PGS}$
M_{AGS}
=0
=1
CALL AGSAG TO ADVANCE X_{AGS} THROUGH TWO SECONDS ADD $\Delta\overline{V}_{AGS}$
$T_{AGS} = T_{AGS} + 2$
$m = m\left(1 - \frac{|\Delta V_E|}{V_E}\right)$
$W_E = mg$
$I_{PRINT} > 0$
YES
NO
DISPLAY BURN QUANTITIES AND STATE VECTORS FOR THE JUST COMPLETED GUIDANCE CYCLE
INIT
=0
= 1
$\delta T = 2$
INIT = 1
$L_{EN} = 13$
YES
NO
$I_{MDC} \geq 2$
YES
9/6
NO
CALL TRANCO AND ROTATE X_A INTO SELENOGRAPHIC COORDINATES GIVING LANDING SITE $\overline{X}_{LS}$
$K_{TOFF} > 1$
NO
YES
I_{COM}
=3
20/11
≠3
$L_{EN} = 14$
YES
15/9
NO
24/14
35/20

Figure B-55.- Continued.

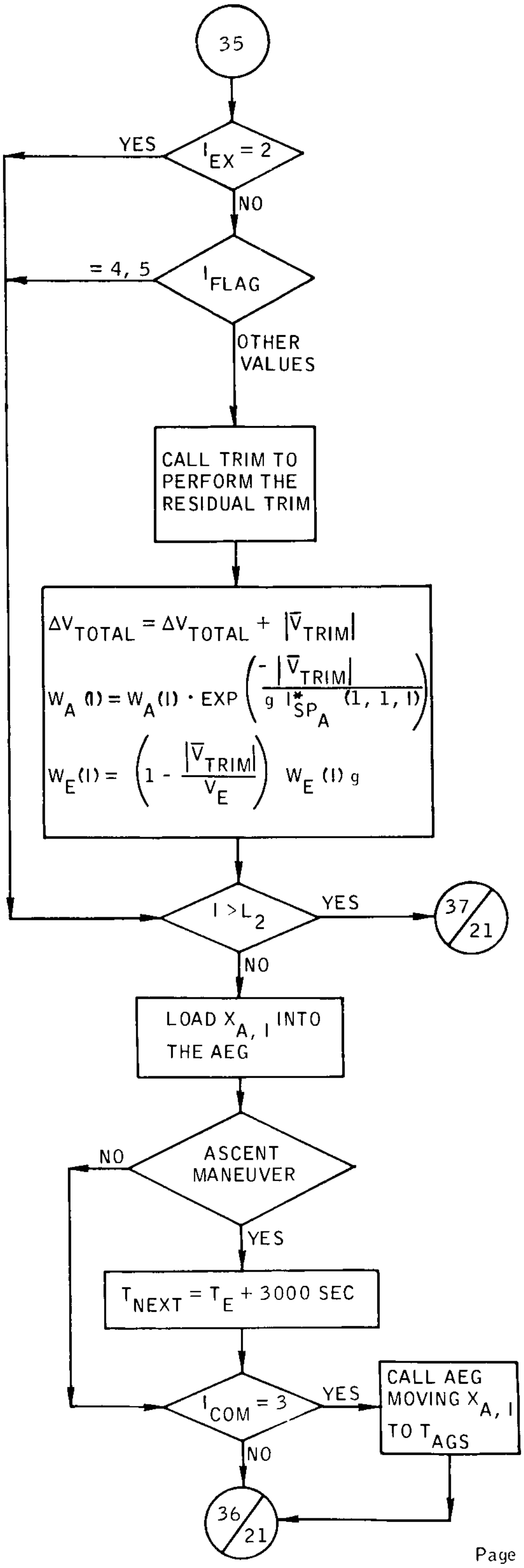

35
YES
$I_{EX} = 2$
NO
= 4, 5
I_{FLAG}
OTHER VALUES
CALL TRIM TO PERFORM THE RESIDUAL TRIM
$\Delta V_{TOTAL} = \Delta V_{TOTAL} + |\bar{V}_{TRIM}|$
$W_A (I) = W_A(I) \cdot EXP \left(\frac{- |\bar{V}_{TRIM}|}{g \; I^*_{SP_A} (1, 1, I)} \right)$
$W_E(I) = \left(1 - \frac{|\bar{V}_{TRIM}|}{V_E} \right) \; W_E (I) g$
$I > L_2$
YES
37/21
NO
LOAD $X_{A, I}$ INTO THE AEG
NO
ASCENT MANEUVER
YES
$T_{NEXT} = T_E + 3000$ SEC
$I_{COM} = 3$
YES
CALL AEG MOVING $X_{A, I}$ TO T_{AGS}
NO
36/21

Figure B-55.- Continued.

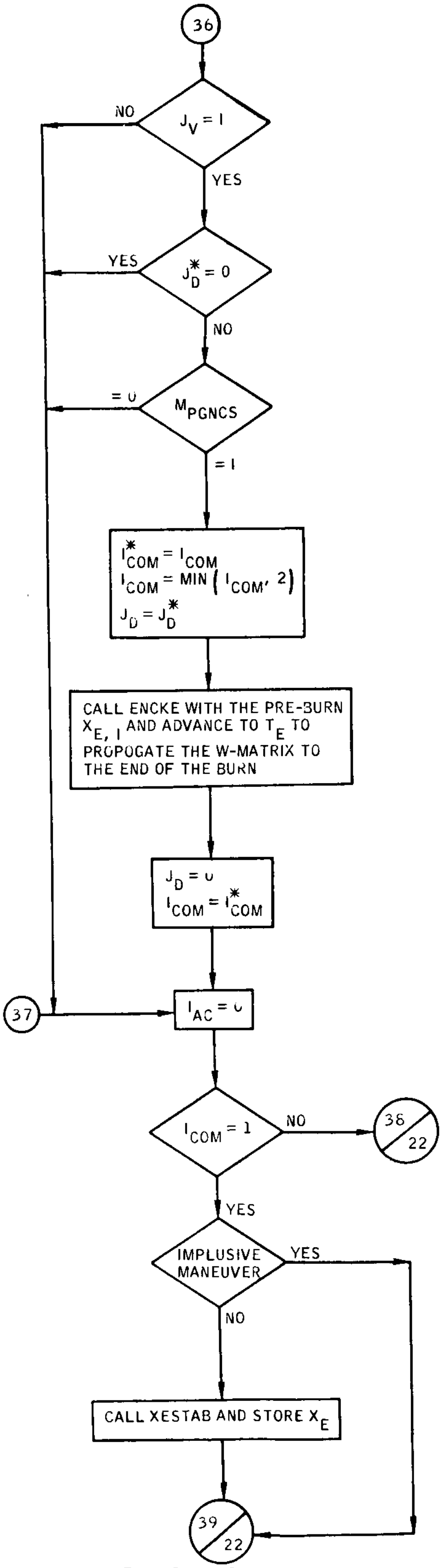


Figure B-55. - Continued.

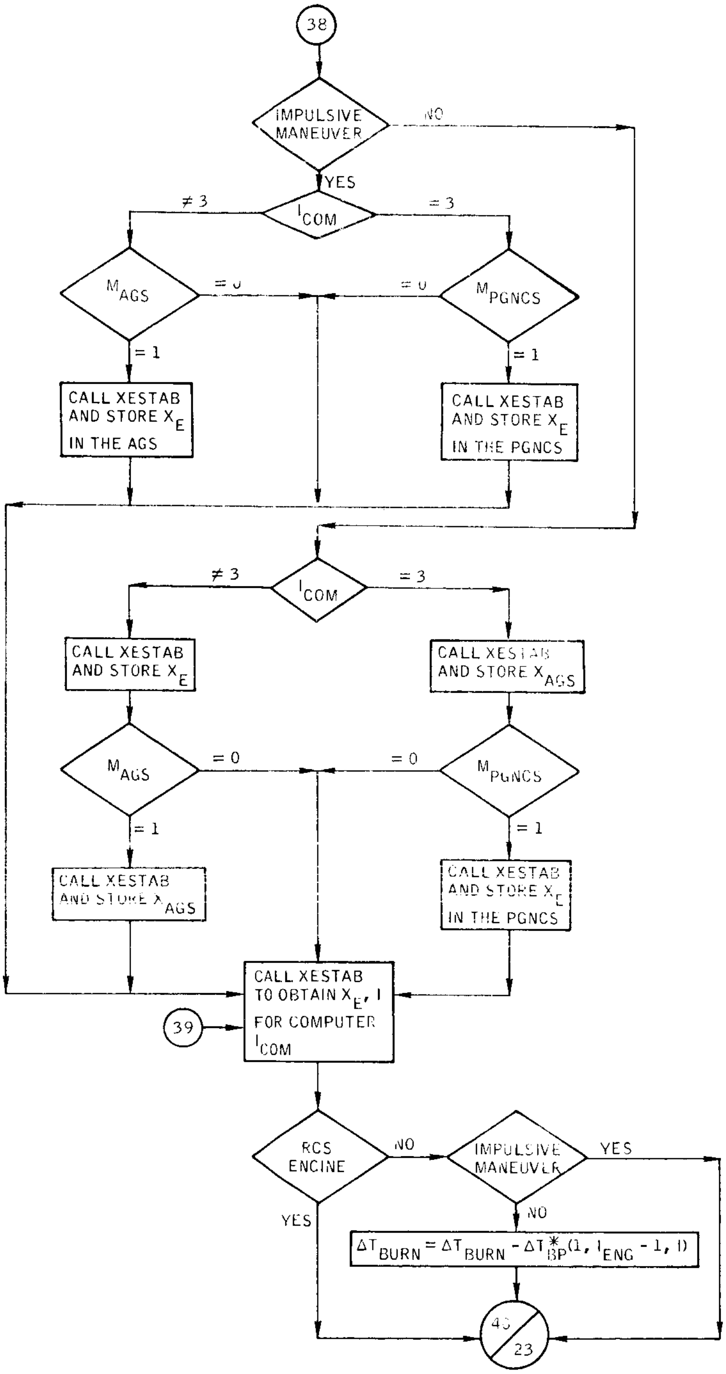

38
IMPULSIVE MANEUVER
NO
YES
I_{COM}
≠ 3
= 3
M_{AGS}
= 0
= 0
M_{PGNCS}
= 1
= 1
CALL XESTAB AND STORE X_E IN THE AGS
CALL XESTAB AND STORE X_E IN THE PGNCS
I_{COM}
≠ 3
= 3
CALL XESTAB AND STORE X_E
CALL XESTAB AND STORE X_{AGS}
M_{AGS}
= 0
= 0
M_{PGNCS}
= 1
= 1
CALL XESTAB AND STORE X_{AGS}
CALL XESTAB AND STORE X_E IN THE PGNCS
CALL XESTAB TO OBTAIN X_E, I FOR COMPUTER I_{COM}
39
RCS ENGINE
NO
IMPULSIVE MANEUVER
YES
YES
NO
$\Delta T_{BURN} = \Delta T_{BURN} - \Delta T^*_{BP}(1, I_{ENG} - 1, I)$
40
23

Figure B-55.- Continued.

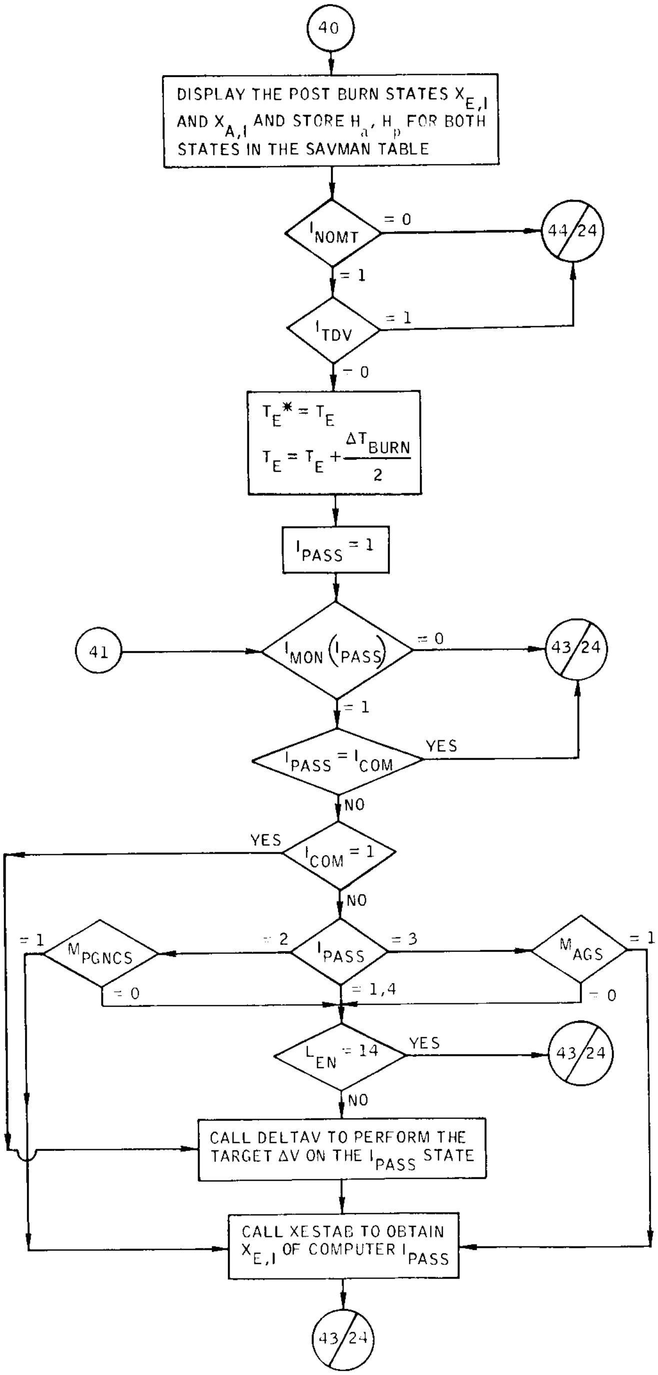

40
DISPLAY THE POST BURN STATES $X_{E,I}$ AND $X_{A,I}$ AND STORE H_a, H_p FOR BOTH STATES IN THE SAVMAN TABLE
I_{NOMT}
= 0
44/24
= 1
I_{TDV}
= 1
= 0
$T_E^* = T_E$
$T_E = T_E + \frac{\Delta T_{BURN}}{2}$
$I_{PASS} = 1$
41
$I_{MON}(I_{PASS})$
= 0
43/24
= 1
$I_{PASS} = I_{COM}$
YES
NO
$I_{COM} = 1$
YES
NO
M_{PGNCS}
= 1
= 0
= 2
I_{PASS}
= 3
= 1,4
M_{AGS}
= 1
= 0
$L_{EN} = 14$
YES
43/24
NO
CALL DELTAV TO PERFORM THE TARGET ΔV ON THE I_{PASS} STATE
CALL XESTAB TO OBTAIN $X_{E,I}$ OF COMPUTER I_{PASS}
43/24

Figure B-55. - Continued.

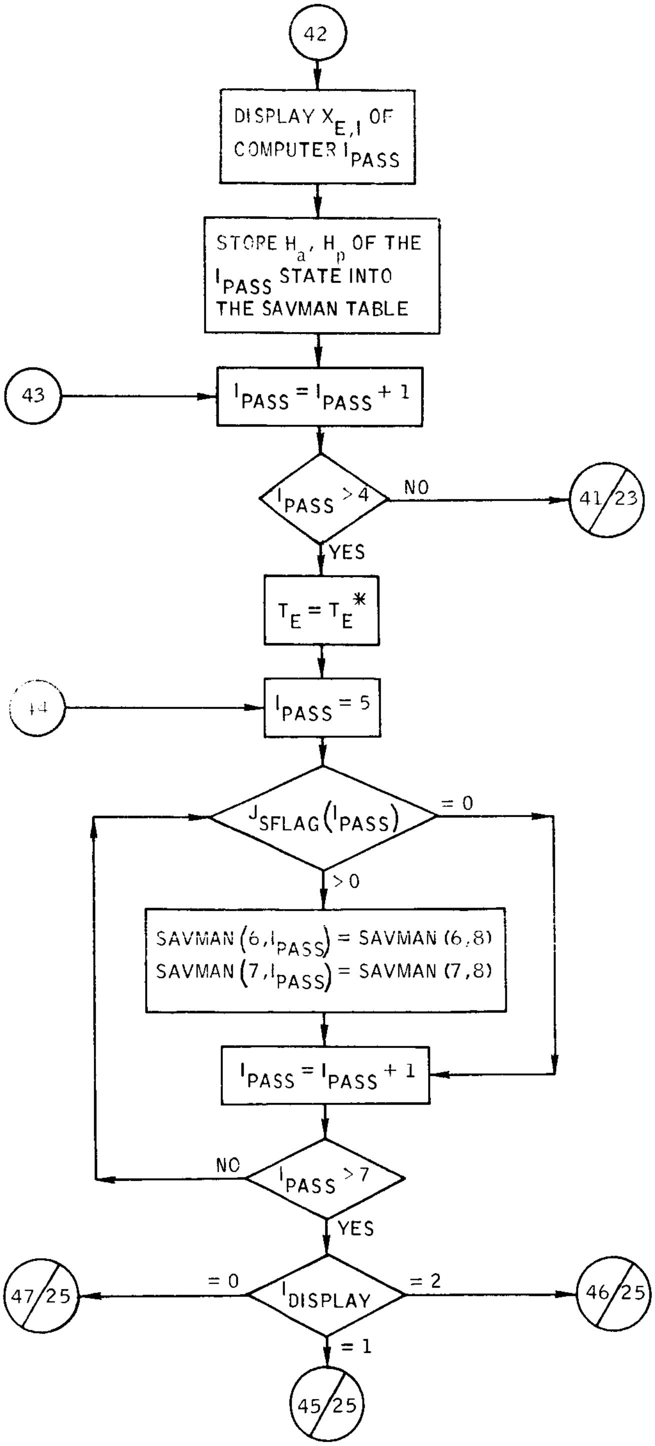
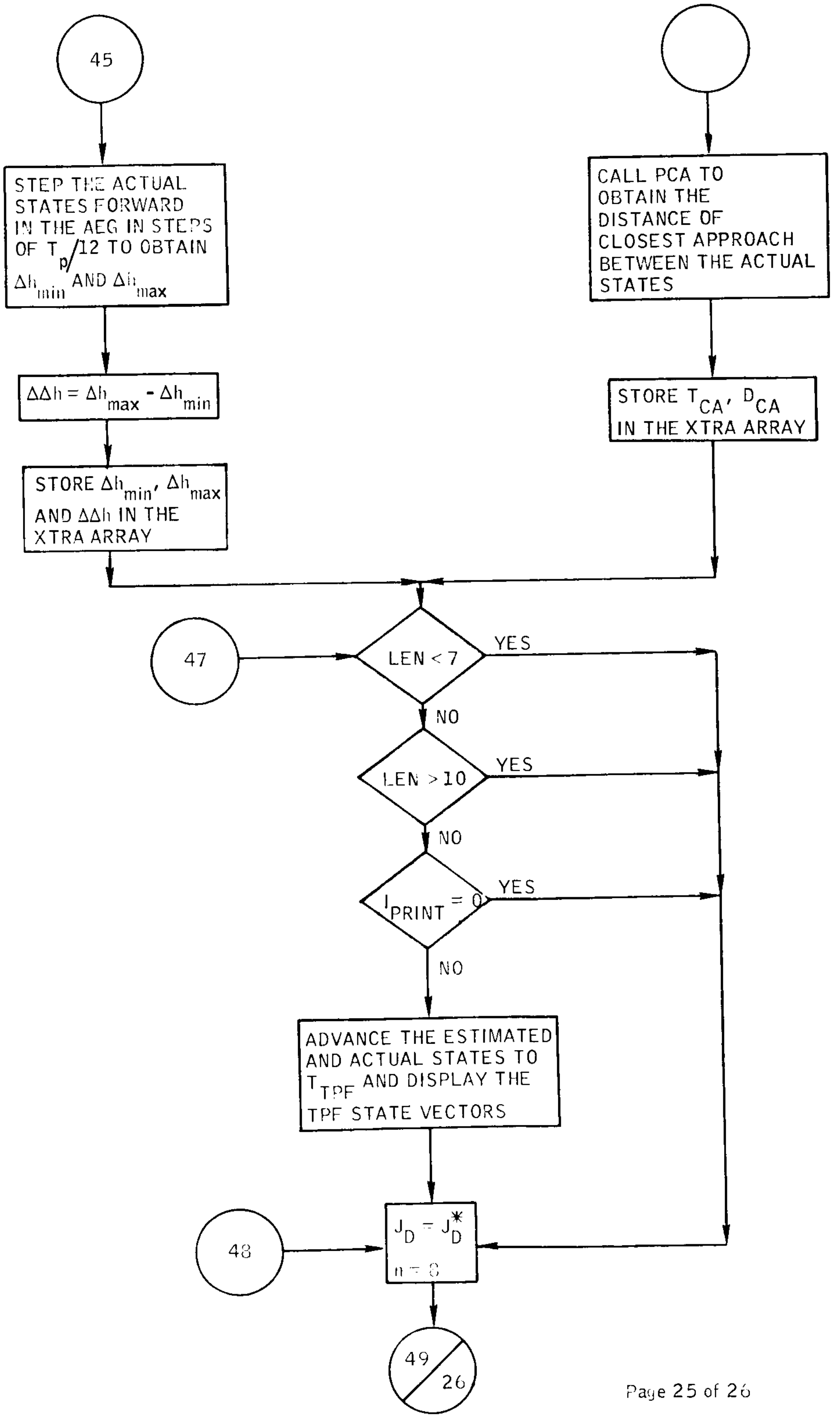
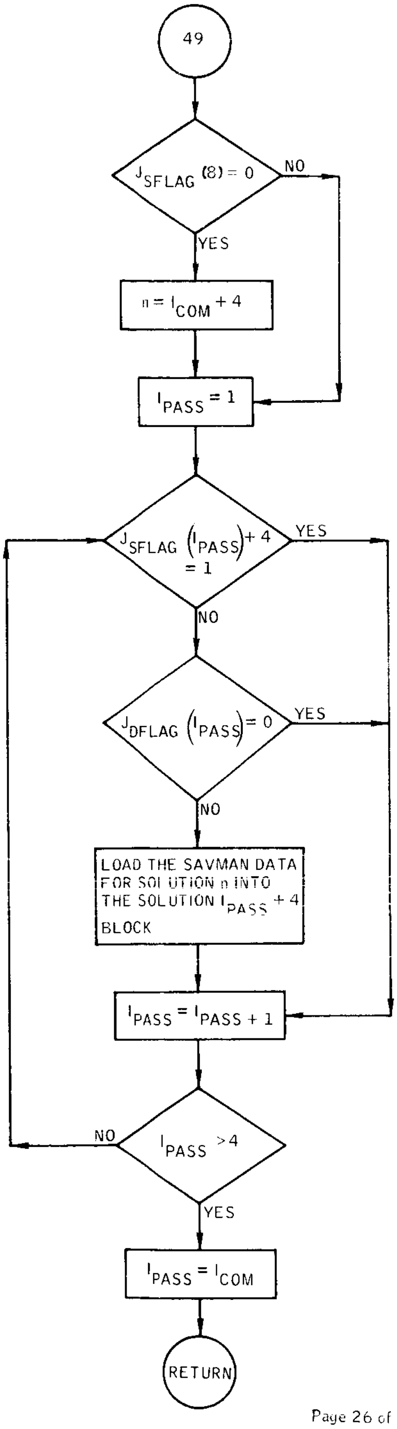

42
DISPLAY $X_{E,I}$ OF COMPUTER I_{PASS}
STORE H_a, H_p OF THE I_{PASS} STATE INTO THE SAVMAN TABLE
43
$I_{PASS} = I_{PASS} + 1$
$I_{PASS} > 4$
NO
41/23
YES
$T_E = T_E^*$
44
$I_{PASS} = 5$
$J_{SFLAG}(I_{PASS})$
= 0
> 0
SAVMAN $(6, I_{PASS})$ = SAVMAN (6,8)
SAVMAN $(7, I_{PASS})$ = SAVMAN (7,8)
$I_{PASS} = I_{PASS} + 1$
NO
$I_{PASS} > 7$
YES
$I_{DISPLAY}$
= 0
47/25
= 2
46/25
= 1
45/25

Figure B-55.- Continued.


45
STEP THE ACTUAL STATES FORWARD IN THE AEG IN STEPS OF $T_p/12$ TO OBTAIN Δh_{min} AND Δh_{max}
$\Delta\Delta h = \Delta h_{max} - \Delta h_{min}$
STORE Δh_{min}, Δh_{max} AND $\Delta\Delta h$ IN THE XTRA ARRAY
CALL PCA TO OBTAIN THE DISTANCE OF CLOSEST APPROACH BETWEEN THE ACTUAL STATES
STORE T_{CA}, D_{CA} IN THE XTRA ARRAY
47
LEN < 7
YES
NO
LEN > 10
YES
NO
$I_{PRINT} = 0$
YES
NO
ADVANCE THE ESTIMATED AND ACTUAL STATES TO T_{TPF} AND DISPLAY THE TPF STATE VECTORS
48
$J_D = J_D^*$
$n = 8$
49
26

Page 25 of 26

Figure B-55.- Continued.


49
J_{SFLAG} (8) = 0
NO
YES
$n = I_{COM} + 4$
$I_{PASS} = 1$
$J_{SFLAG}\left(I_{PASS}\right) + 4 = 1$
YES
NO
$J_{DFLAG}\left(I_{PASS}\right) = 0$
YES
NO
LOAD THE SAVMAN DATA FOR SOLUTION n INTO THE SOLUTION $I_{PASS} + 4$ BLOCK
$I_{PASS} = I_{PASS} + 1$
$I_{PASS} > 4$
NO
YES
$I_{PASS} = I_{COM}$
RETURN

Page 26 of 26

Figure B-55.- Concluded.

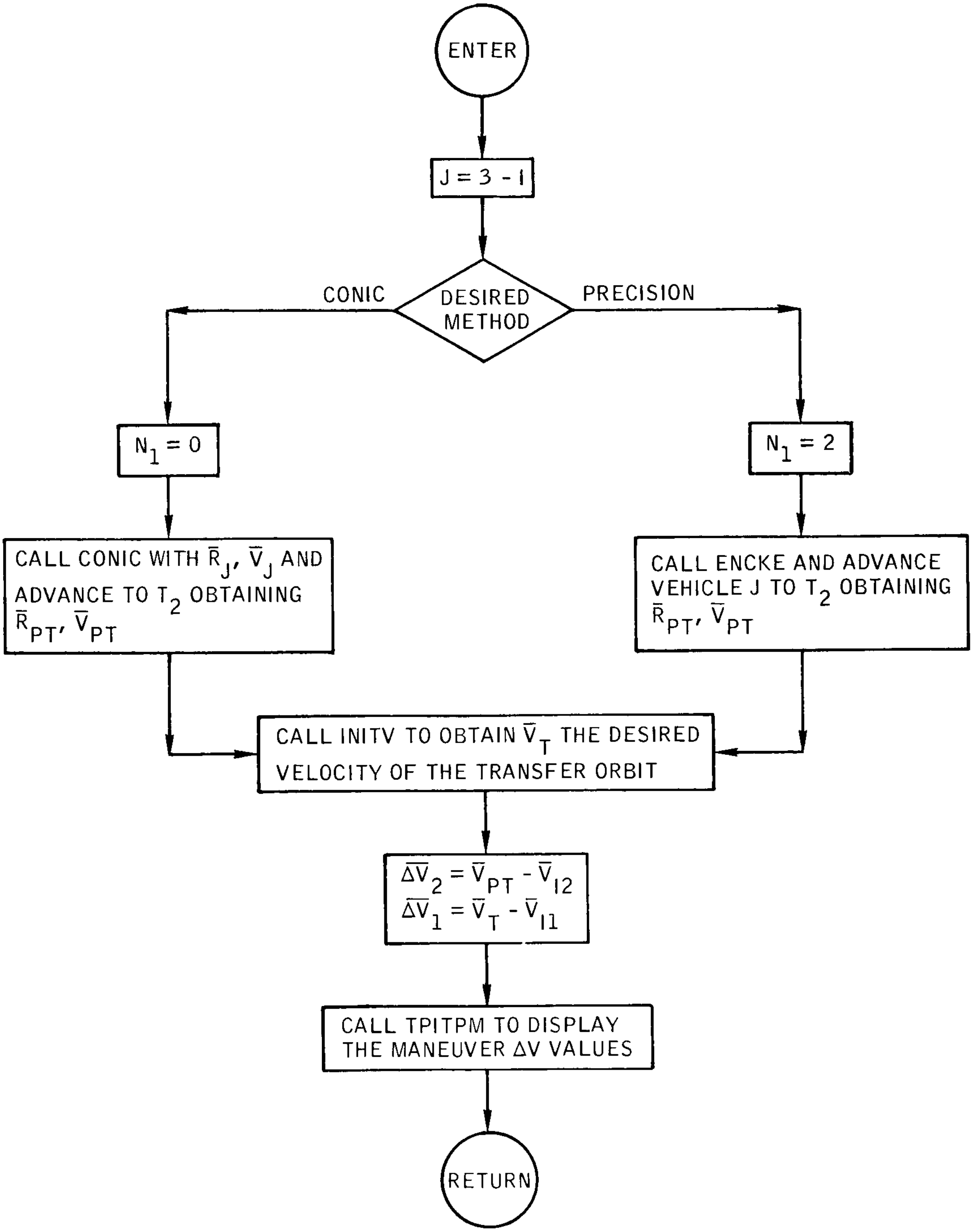

ENTER
J = 3 - I
DESIRED METHOD
CONIC
PRECISION
$N_1 = 0$
$N_1 = 2$
CALL CONIC WITH $\bar{R}_J$, $\bar{V}_J$ AND ADVANCE TO T_2 OBTAINING $\bar{R}_{PT}$, $\bar{V}_{PT}$
CALL ENCKE AND ADVANCE VEHICLE J TO T_2 OBTAINING $\bar{R}_{PT}$, $\bar{V}_{PT}$
CALL INITV TO OBTAIN $\bar{V}_T$ THE DESIRED VELOCITY OF THE TRANSFER ORBIT
$\overline{\Delta V}_2 = \bar{V}_{PT} - \bar{V}_{I2}$
$\overline{\Delta V}_1 = \bar{V}_T - \bar{V}_{I1}$
CALL TPITPM TO DISPLAY THE MANEUVER ΔV VALUES
RETURN

Figure B-56.- Flow chart of subroutine MIDCRS.

Page 1 of 1

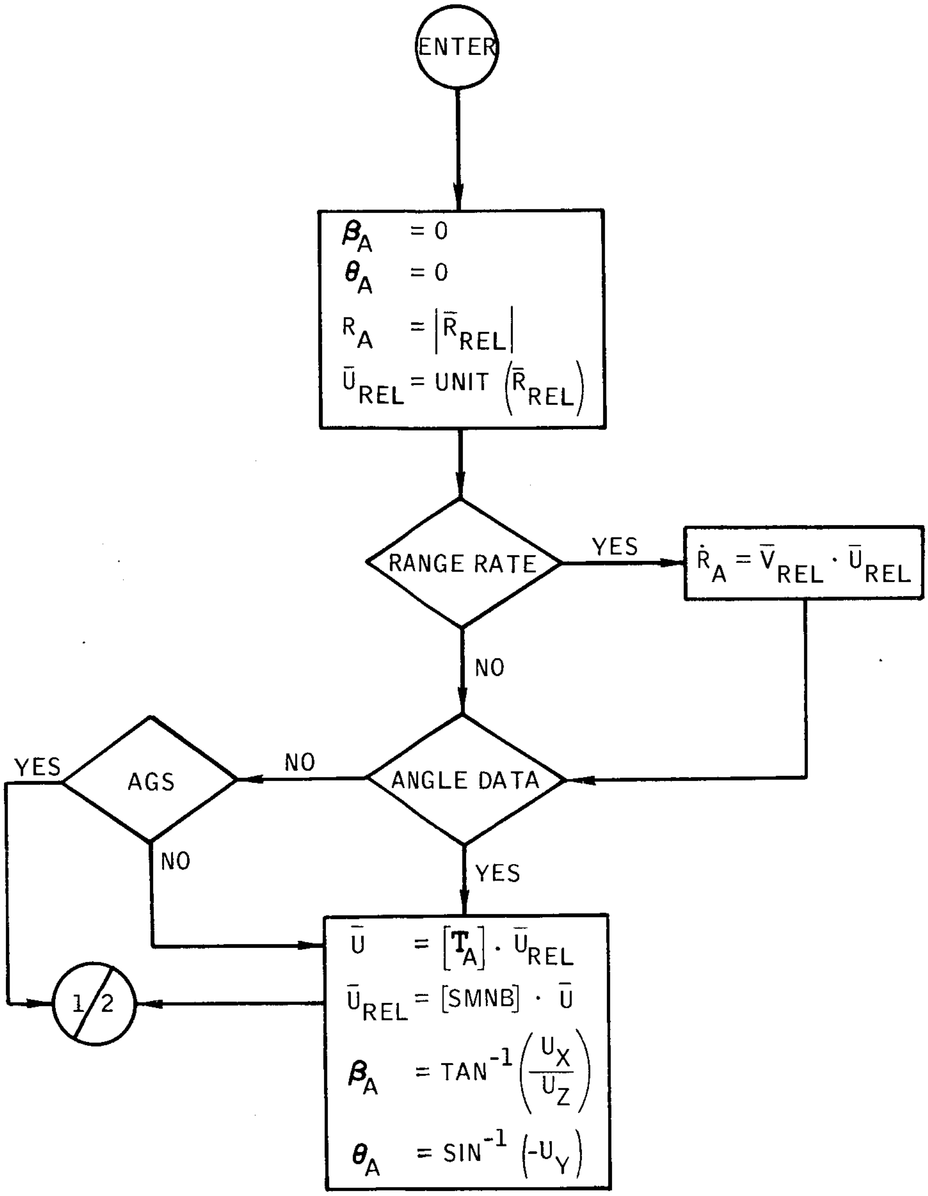
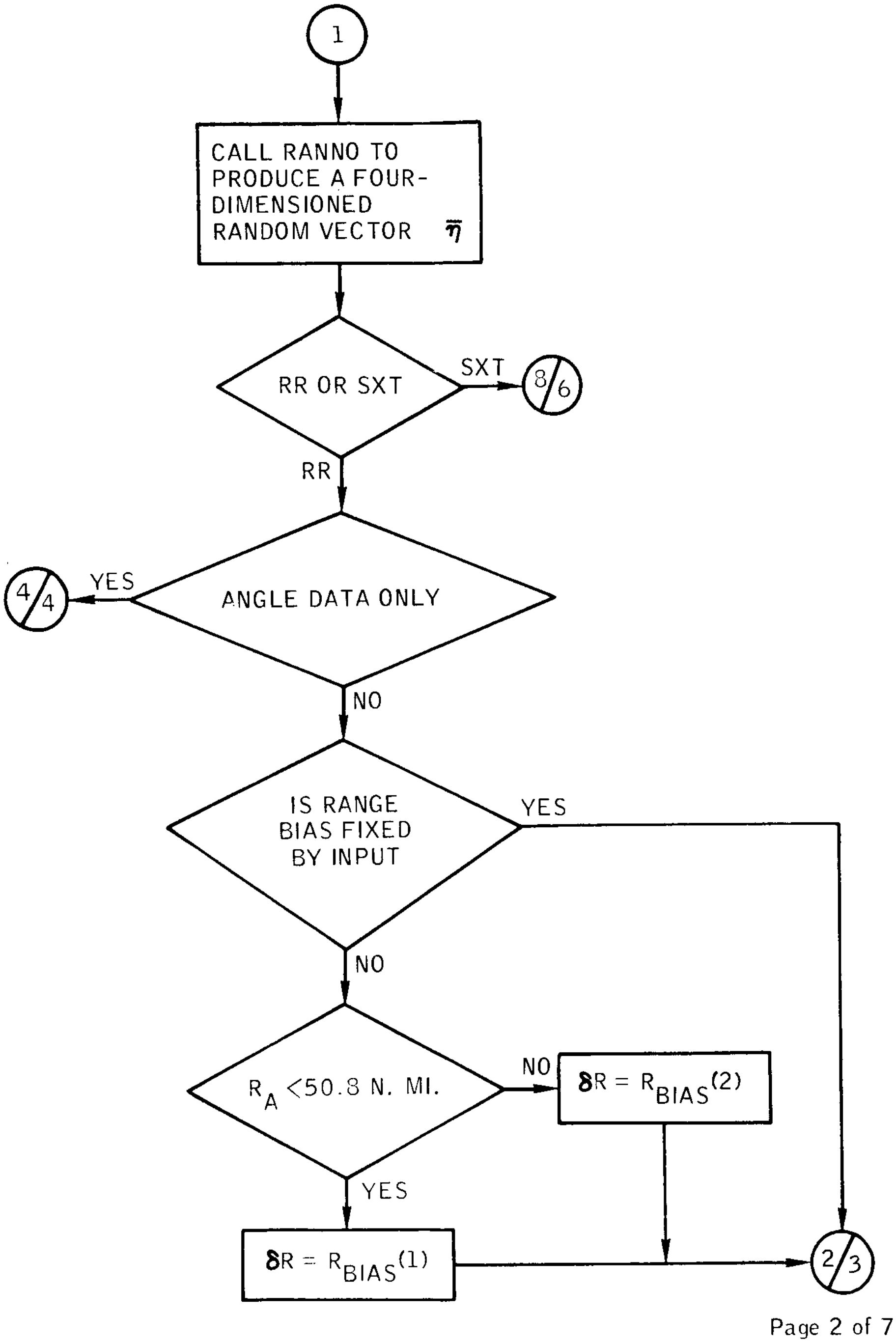

ENTER
$\beta_A = 0$
$\theta_A = 0$
$R_A = |\bar{R}_{REL}|$
$\bar{U}_{REL} = UNIT(\bar{R}_{REL})$
RANGE RATE
YES
$\dot{R}_A = \bar{V}_{REL} \cdot \bar{U}_{REL}$
NO
ANGLE DATA
NO
AGS
YES
NO
YES
$\bar{U} = [T_A] \cdot \bar{U}_{REL}$
$\bar{U}_{REL} = [SMNB] \cdot \bar{U}$
$\beta_A = TAN^{-1}\left(\frac{U_X}{U_Z}\right)$
$\theta_A = SIN^{-1}(-U_Y)$
1/2

Figure B-57.- Flow chart of subroutine OBS.

Page 1 of 7



Page 2 of 7

Figure B-57.- Continued.

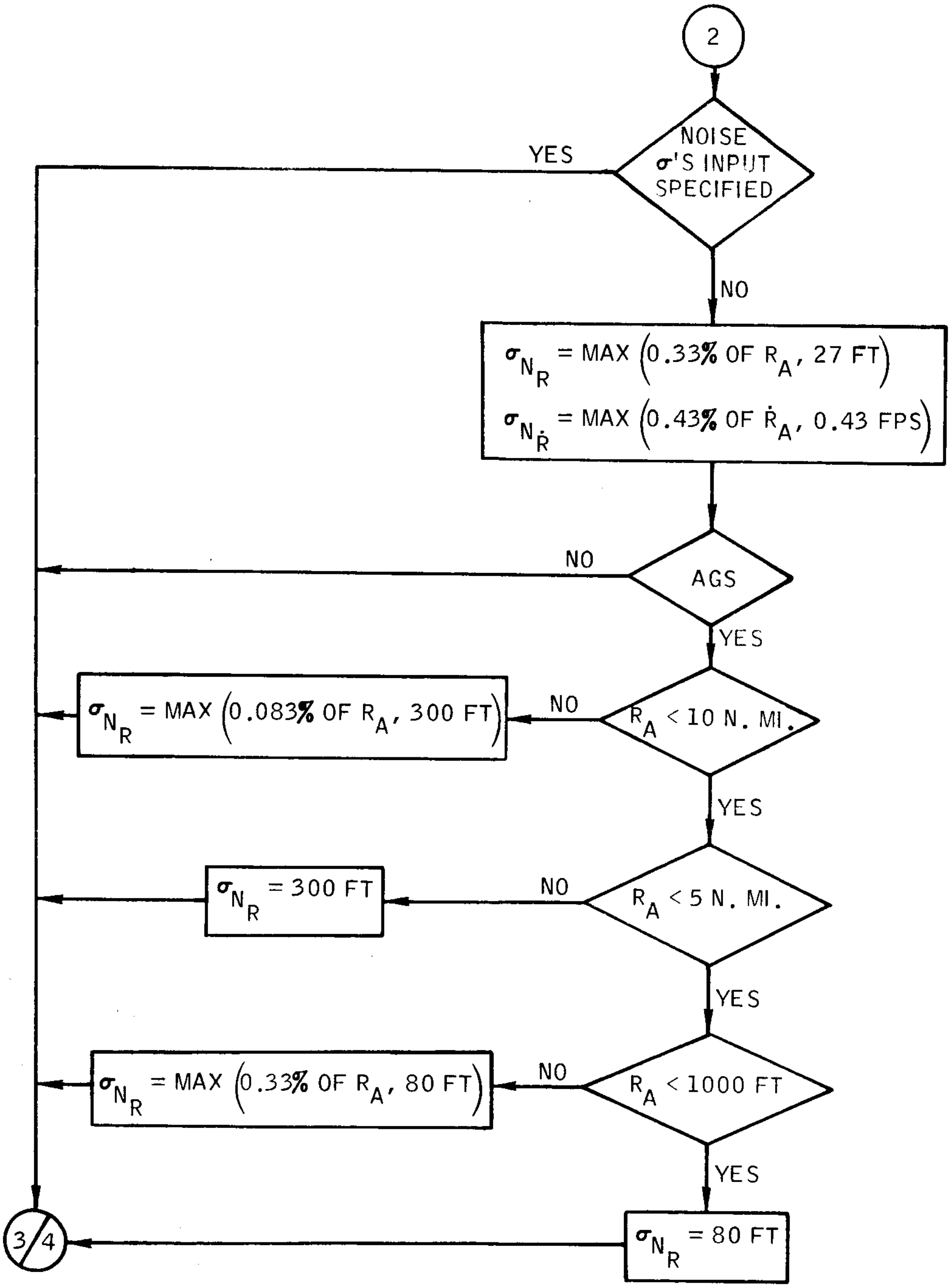
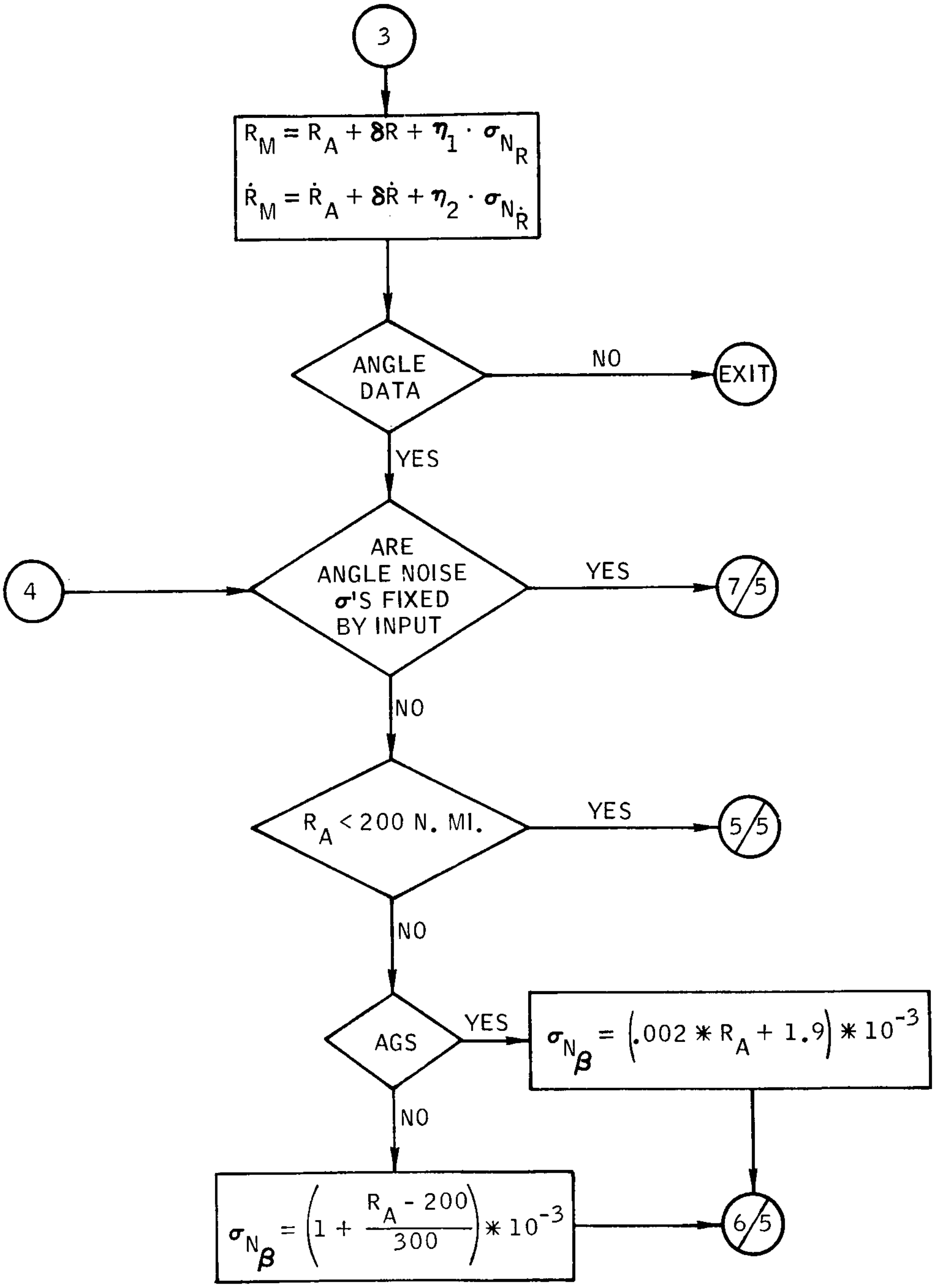
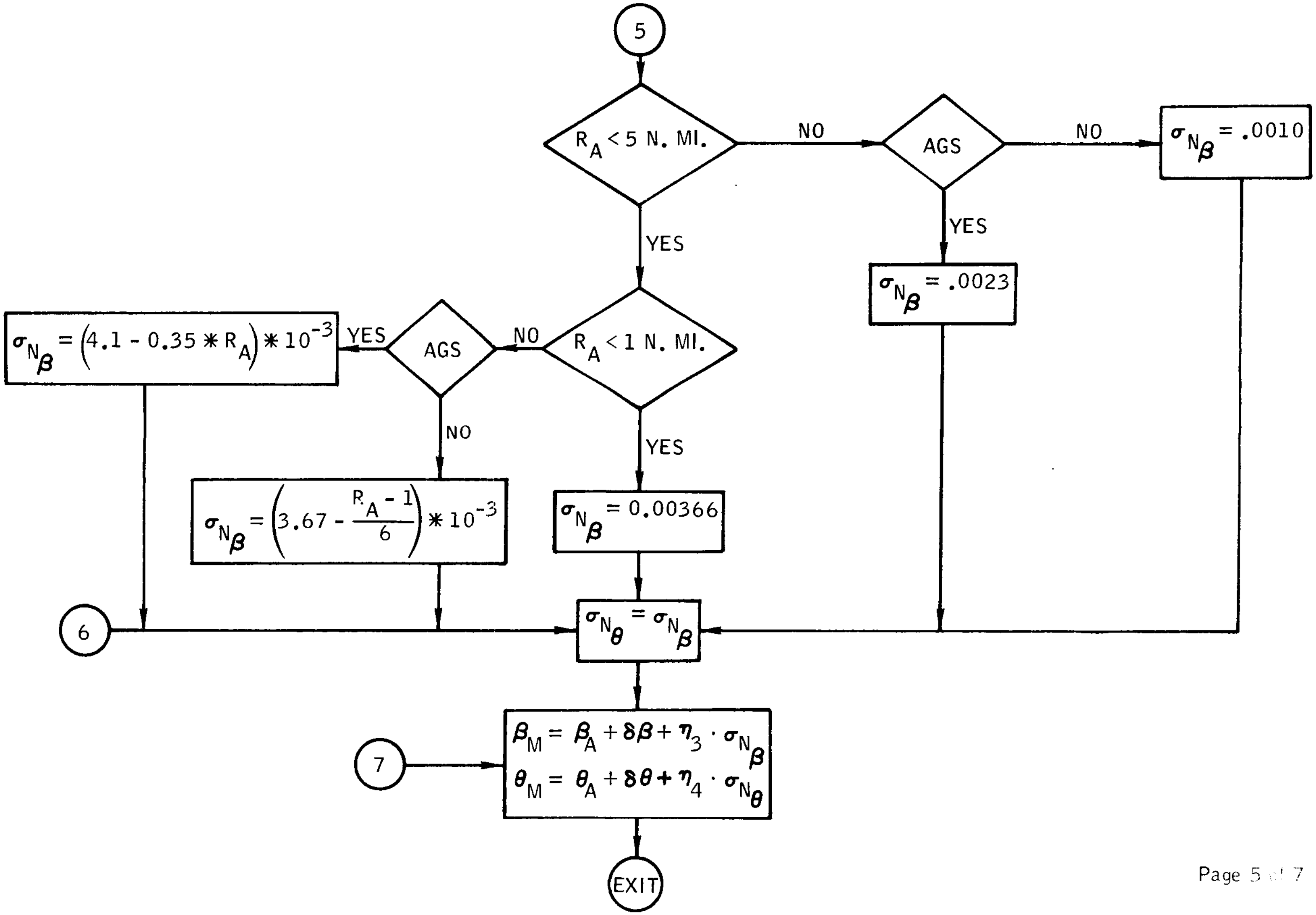

2
NOISE σ'S INPUT SPECIFIED
YES
NO
σ_{N_R} = MAX (0.33% OF R_A, 27 FT)
$\sigma_{N_{\dot{R}}}$ = MAX (0.43% OF $\dot{R}_A$, 0.43 FPS)
AGS
NO
YES
R_A < 10 N. MI.
NO
σ_{N_R} = MAX (0.083% OF R_A, 300 FT)
YES
R_A < 5 N. MI.
NO
σ_{N_R} = 300 FT
YES
R_A < 1000 FT
NO
σ_{N_R} = MAX (0.33% OF R_A, 80 FT)
YES
σ_{N_R} = 80 FT
3/4

Figure B-57.- Continued.


3
$R_M = R_A + \delta R + \eta_1 \cdot \sigma_{N_R}$
$\dot{R}_M = \dot{R}_A + \delta \dot{R} + \eta_2 \cdot \sigma_{N_{\dot{R}}}$
ANGLE DATA
NO
EXIT
YES
4
ARE ANGLE NOISE σ'S FIXED BY INPUT
YES
7/5
NO
$R_A < 200$ N. MI.
YES
5/5
NO
AGS
YES
$\sigma_{N_\beta} = \left(.002 * R_A + 1.9\right) * 10^{-3}$
NO
$\sigma_{N_\beta} = \left(1 + \frac{R_A - 200}{300}\right) * 10^{-3}$
6/5

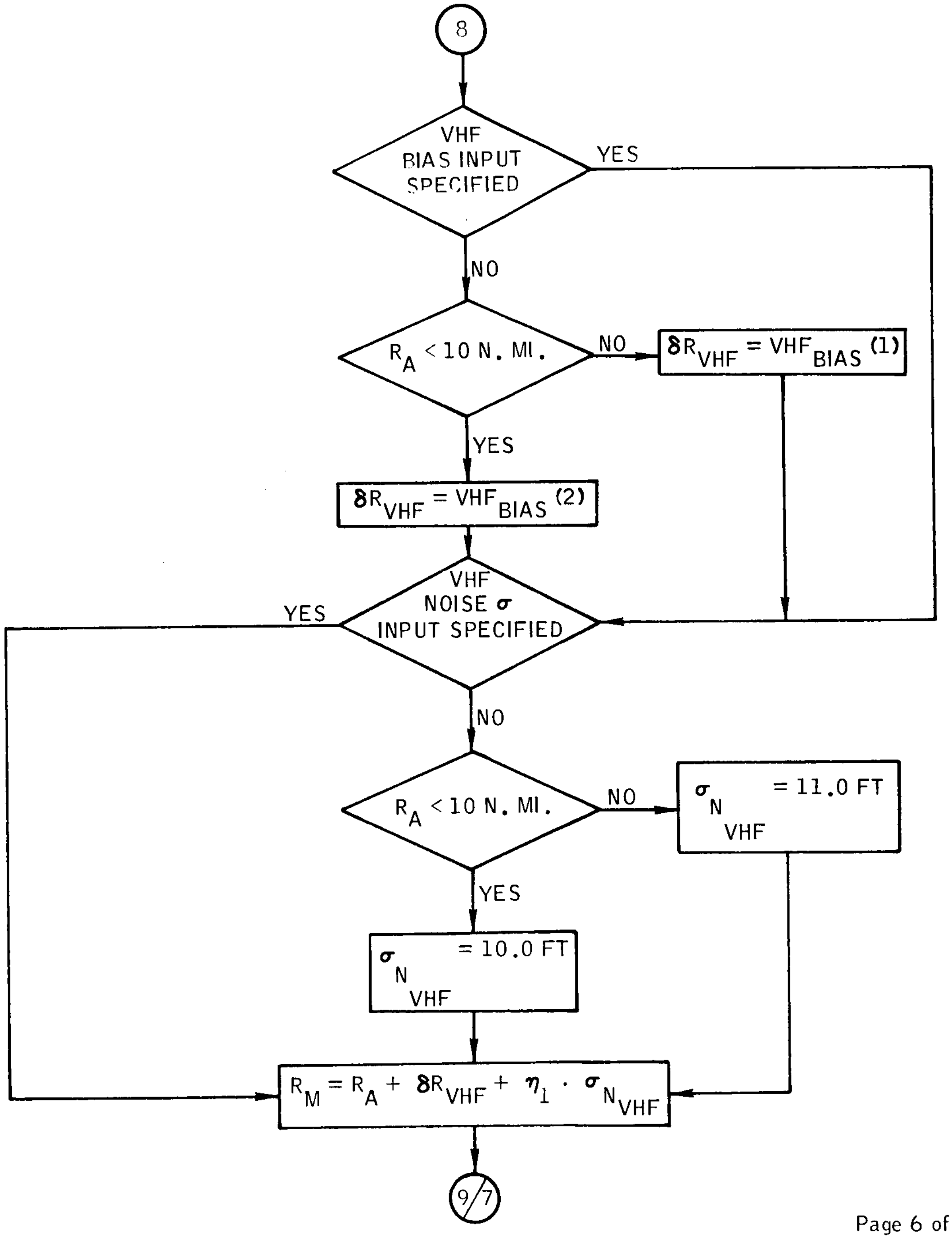
Page 4 of 7

Figure B-57.- Continued.


5
$R_A < 5$ N. MI.
NO
AGS
NO
$\sigma_{N_\beta} = .0010$
YES
$\sigma_{N_\beta} = .0023$
YES
$R_A < 1$ N. MI.
NO
AGS
YES
$\sigma_{N_\beta} = \left(4.1 - 0.35 * R_A\right) * 10^{-3}$
NO
$\sigma_{N_\beta} = \left(3.67 - \frac{R_A - 1}{6}\right) * 10^{-3}$
YES
$\sigma_{N_\beta} = 0.00366$
6
$\sigma_{N_\theta} = \sigma_{N_\beta}$
7
$\beta_M = \beta_A + \delta\beta + \eta_3 \cdot \sigma_{N_\beta}$
$\theta_M = \theta_A + \delta\theta + \eta_4 \cdot \sigma_{N_\theta}$
EXIT

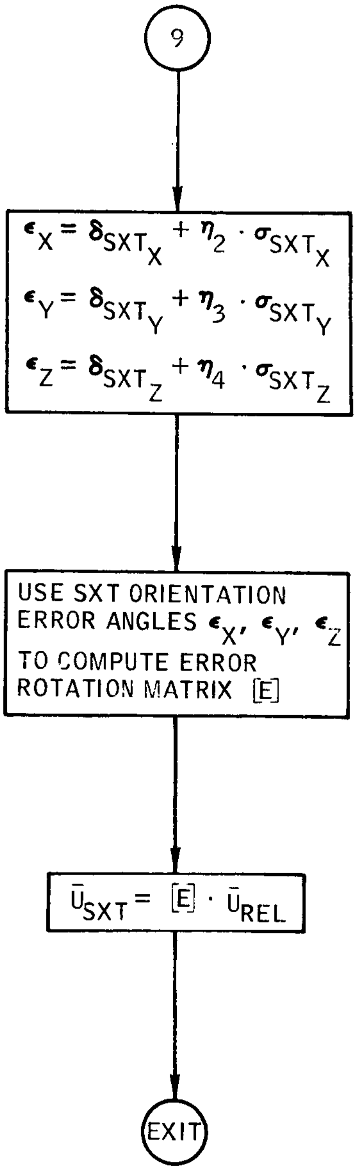
Page 5 of 7

Figure B-57.- Continued.


8
VHF BIAS INPUT SPECIFIED
YES
NO
$R_A < 10$ N. MI.
NO
$\delta R_{VHF} = VHF_{BIAS}$ (1)
YES
$\delta R_{VHF} = VHF_{BIAS}$ (2)
VHF NOISE σ INPUT SPECIFIED
YES
NO
$R_A < 10$ N. MI.
NO
$\sigma_{N_{VHF}} = 11.0$ FT
YES
$\sigma_{N_{VHF}} = 10.0$ FT
$R_M = R_A + \delta R_{VHF} + \eta_1 \cdot \sigma_{N_{VHF}}$
9/7

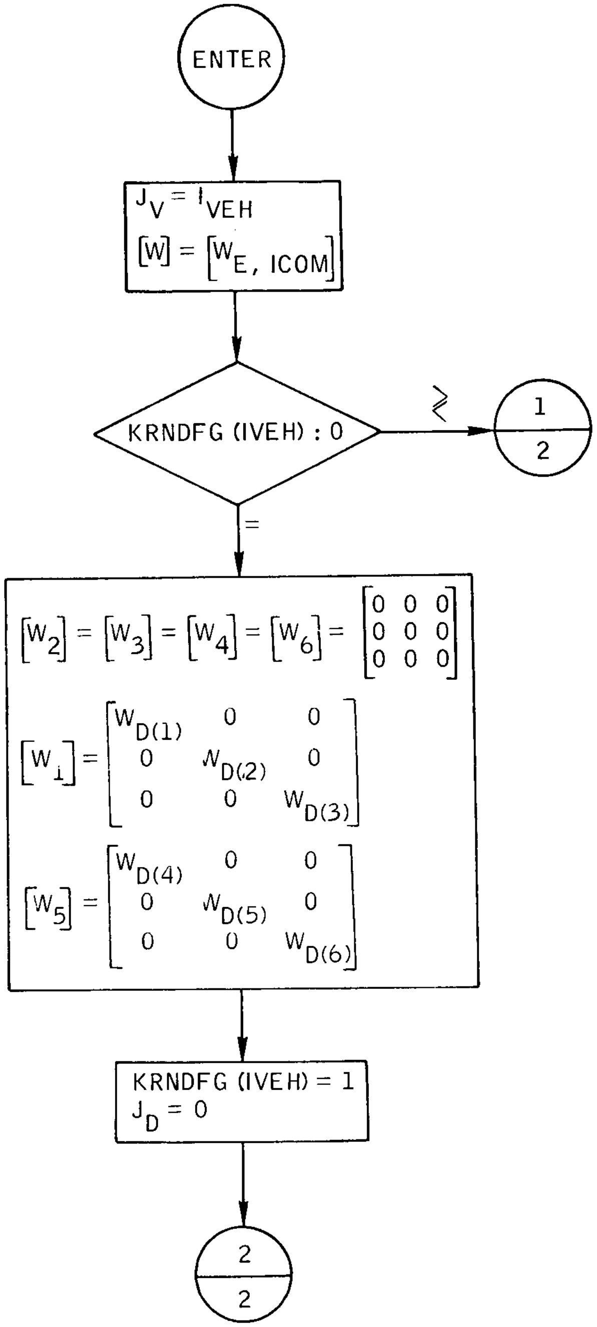
Page 6 of 7

Figure B-57.- Continued.


9
$\epsilon_X = \delta_{SXT_X} + \eta_2 \cdot \sigma_{SXT_X}$
$\epsilon_Y = \delta_{SXT_Y} + \eta_3 \cdot \sigma_{SXT_Y}$
$\epsilon_Z = \delta_{SXT_Z} + \eta_4 \cdot \sigma_{SXT_Z}$
USE SXT ORIENTATION ERROR ANGLES ϵ_X, ϵ_Y, ϵ_Z TO COMPUTE ERROR ROTATION MATRIX [E]
$\bar{U}_{SXT} = [E] \cdot \bar{U}_{REL}$
EXIT

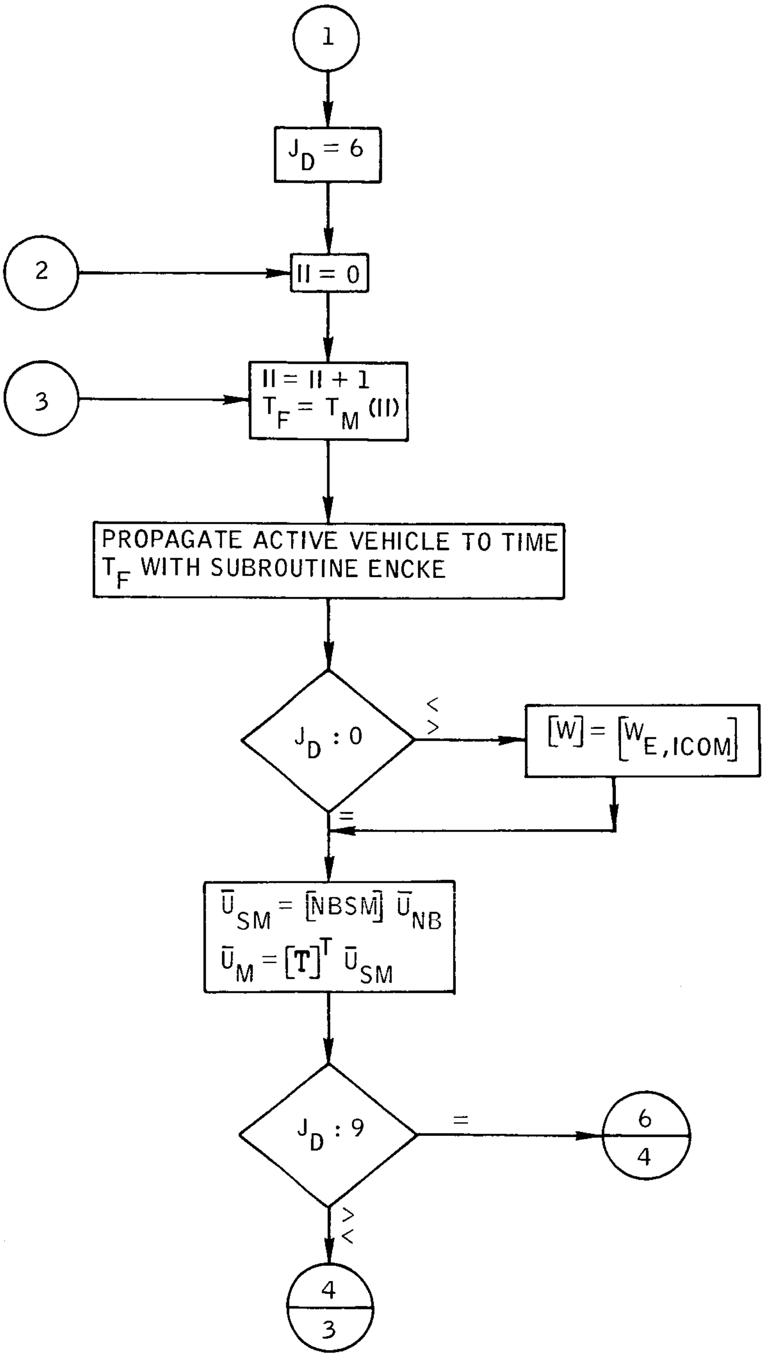
Page 7 of 7

Figure B-57.- Concluded.


ENTER
$J_V = I_{VEH}$
$[W] = [W_{E,\ ICOM}]$
KRNDFG (IVEH) : 0
$\gtrless$
1
2
=
$[W_2] = [W_3] = [W_4] = [W_6] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
$[W_1] = \begin{bmatrix} W_{D(1)} & 0 & 0 \\ 0 & W_{D(2)} & 0 \\ 0 & 0 & W_{D(3)} \end{bmatrix}$
$[W_5] = \begin{bmatrix} W_{D(4)} & 0 & 0 \\ 0 & W_{D(5)} & 0 \\ 0 & 0 & W_{D(6)} \end{bmatrix}$
KRNDFG (IVEH) = 1
$J_D = 0$
2
2

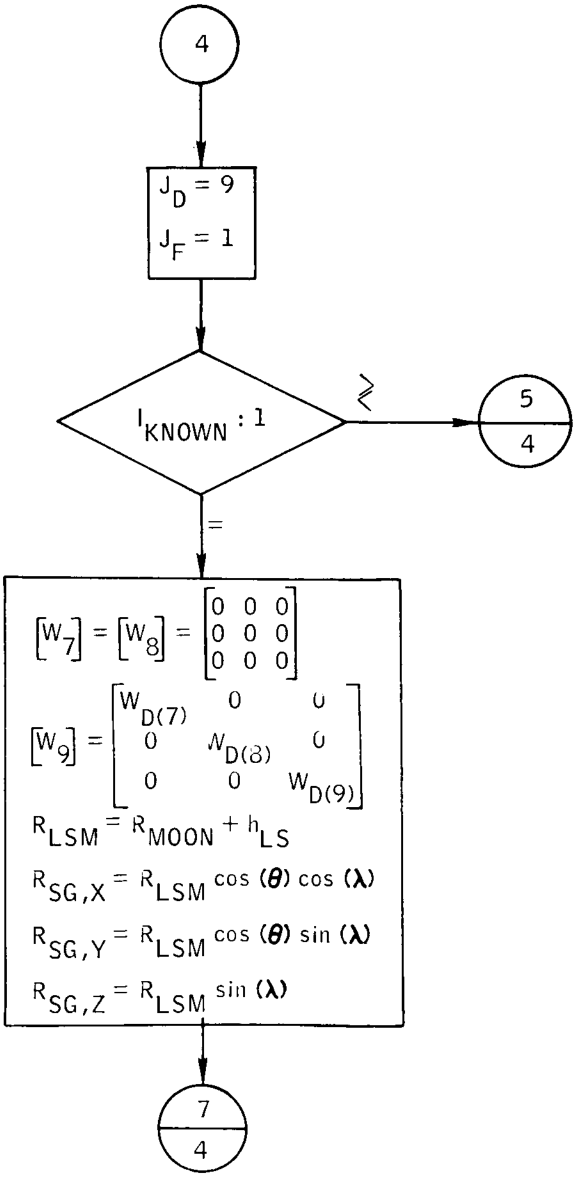
Page 1 of 7

Figure B-58.- Flow chart of subroutine ONR.


1
$J_D = 6$
2
$II = 0$
3
$II = II + 1$
$T_F = T_M\ (II)$
PROPAGATE ACTIVE VEHICLE TO TIME T_F WITH SUBROUTINE ENCKE
$J_D : 0$
<
>
=
$[W] = [W_{E,ICOM}]$
$\bar{U}_{SM} = [NBSM]\ \bar{U}_{NB}$
$\bar{U}_M = [T]^T\ \bar{U}_{SM}$
$J_D : 9$
=
6
4
>
<
4
3

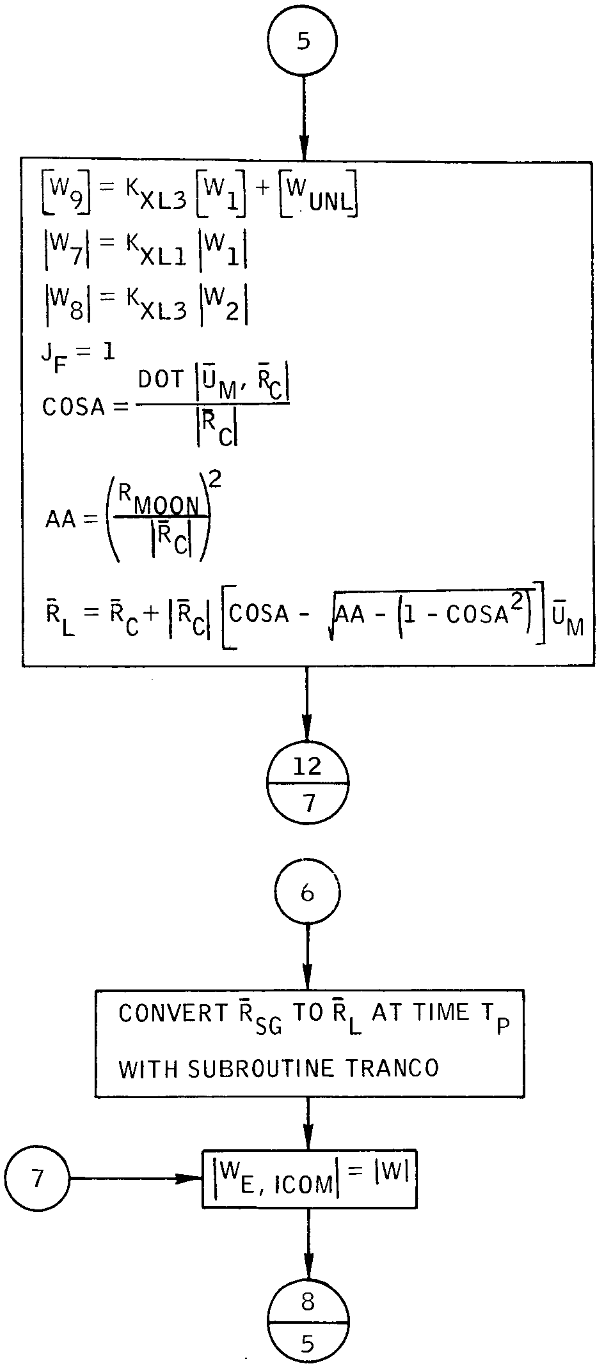
Page 2 of 7

Figure B-58.- Continued.


4
$J_D = 9$
$J_F = 1$
$I_{KNOWN} : 1$
$\gtrless$
5
4
=
$[W_7] = [W_8] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
$[W_9] = \begin{bmatrix} W_{D(7)} & 0 & 0 \\ 0 & W_{D(8)} & 0 \\ 0 & 0 & W_{D(9)} \end{bmatrix}$
$R_{LSM} = R_{MOON} + h_{LS}$
$R_{SG,X} = R_{LSM} \cos(\theta) \cos(\lambda)$
$R_{SG,Y} = R_{LSM} \cos(\theta) \sin(\lambda)$
$R_{SG,Z} = R_{LSM} \sin(\lambda)$
7
4

Page 3 of 7

Figure B-58.- Continued.



$$\left[W_9\right] = K_{XL3}\left[W_1\right] + \left[W_{UNL}\right]$$

$$\left|W_7\right| = K_{XL1}\left|W_1\right|$$

$$\left|W_8\right| = K_{XL3}\left|W_2\right|$$

$$J_F = 1$$

$$COSA = \frac{DOT\left|\bar{U}_M, \bar{R}_C\right|}{\left|\bar{R}_C\right|}$$

$$AA = \left(\frac{R_{MOON}}{\left|\bar{R}_C\right|}\right)^2$$

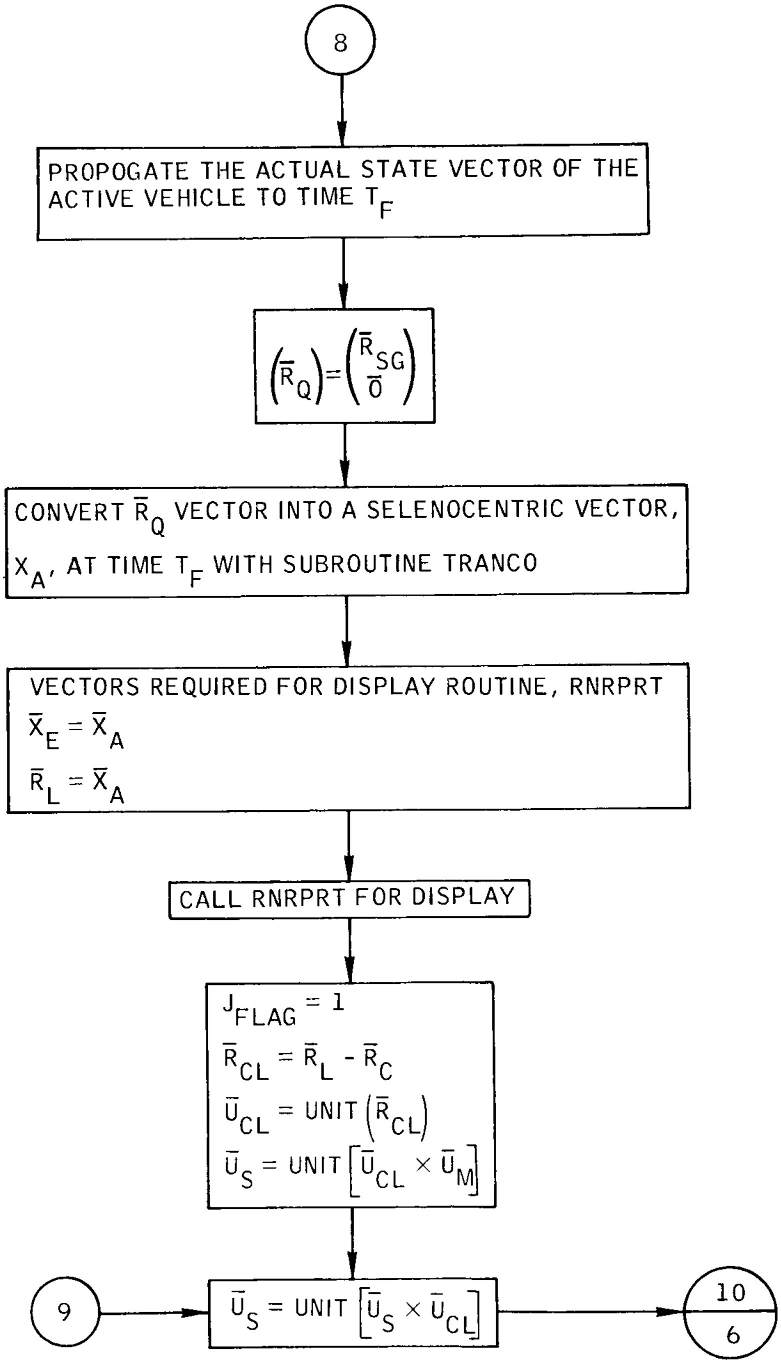
$$\bar{R}_L = \bar{R}_C + \left|\bar{R}_C\right| \left[COSA - \sqrt{AA - \left(1 - COSA^2\right)}\right] \bar{U}_M$$

CONVERT $\bar{R}_{SG}$ TO $\bar{R}_L$ AT TIME T_P WITH SUBROUTINE TRANCO

$$\left|W_{E,\,ICOM}\right| = \left|W\right|$$

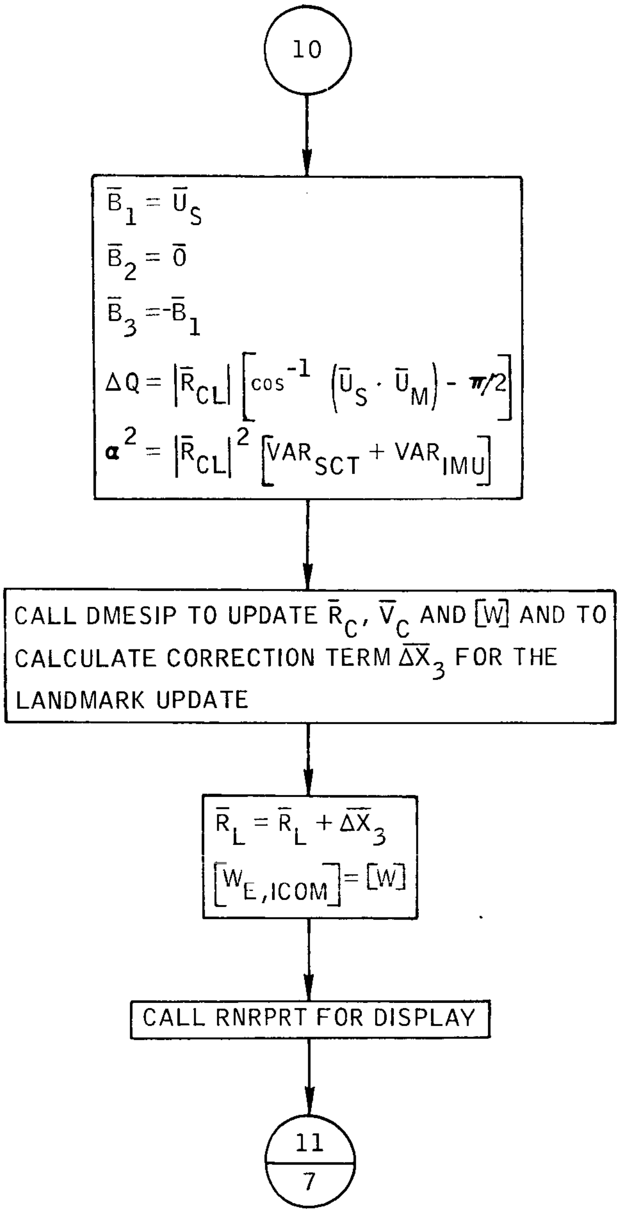
Page 4 of 7

Figure B-58.- Continued.


8
PROPOGATE THE ACTUAL STATE VECTOR OF THE ACTIVE VEHICLE TO TIME T_F
$\left(\bar{R}_Q\right)=\left(\begin{matrix}\bar{R}_{SG}\\ \bar{0}\end{matrix}\right)$
CONVERT $\bar{R}_Q$ VECTOR INTO A SELENOCENTRIC VECTOR, X_A, AT TIME T_F WITH SUBROUTINE TRANCO
VECTORS REQUIRED FOR DISPLAY ROUTINE, RNRPRT
$\bar{X}_E = \bar{X}_A$
$\bar{R}_L = \bar{X}_A$
CALL RNRPRT FOR DISPLAY
$J_{FLAG} = 1$
$\bar{R}_{CL} = \bar{R}_L - \bar{R}_C$
$\bar{U}_{CL} = \text{UNIT}\left(\bar{R}_{CL}\right)$
$\bar{U}_S = \text{UNIT}\left[\bar{U}_{CL} \times \bar{U}_M\right]$
9
$\bar{U}_S = \text{UNIT}\left[\bar{U}_S \times \bar{U}_{CL}\right]$
10
6

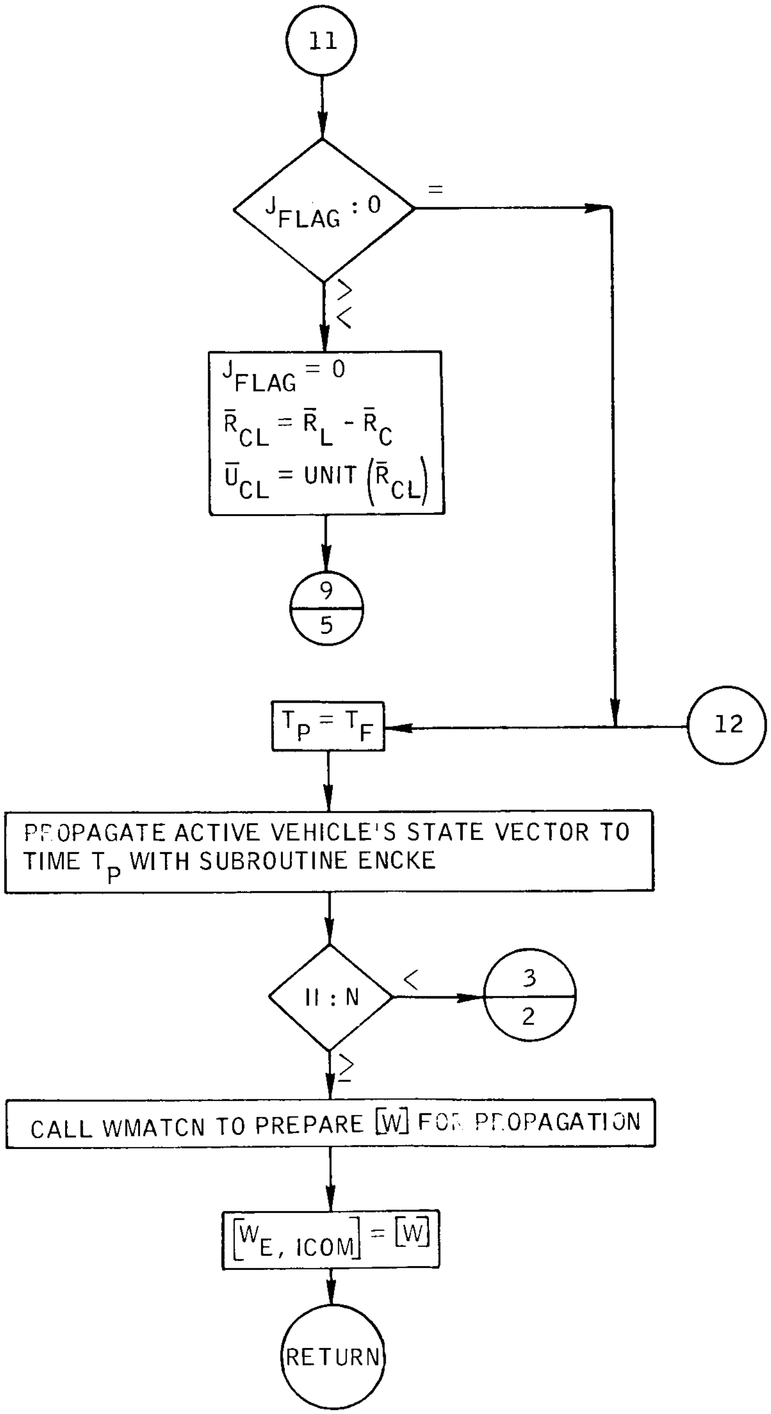
Page 5 of 7

Figure B-58.- Continued.


10
$\bar{B}_1 = \bar{U}_S$
$\bar{B}_2 = \bar{0}$
$\bar{B}_3 = -\bar{B}_1$
$\Delta Q = |\bar{R}_{CL}| \left[\cos^{-1} \left(\bar{U}_S \cdot \bar{U}_M\right) - \pi/2\right]$
$\alpha^2 = |\bar{R}_{CL}|^2 \left[VAR_{SCT} + VAR_{IMU}\right]$
CALL DMESIP TO UPDATE $\bar{R}_C$, $\bar{V}_C$ AND $[W]$ AND TO CALCULATE CORRECTION TERM $\overline{\Delta X}_3$ FOR THE LANDMARK UPDATE
$\bar{R}_L = \bar{R}_L + \overline{\Delta X}_3$
$\left[W_{E,ICOM}\right] = [W]$
CALL RNRPRT FOR DISPLAY
11
7

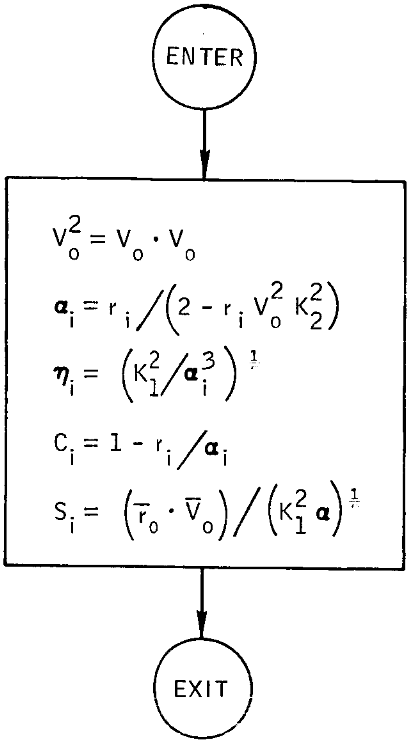
Page 6 of 7

Figure B-58.- Continued.


11
$J_{FLAG} : 0$
=
>
<
$J_{FLAG} = 0$
$\bar{R}_{CL} = \bar{R}_L - \bar{R}_C$
$\bar{U}_{CL} = \text{UNIT}\left(\bar{R}_{CL}\right)$
9
5
$T_P = T_F$
12
PROPAGATE ACTIVE VEHICLE'S STATE VECTOR TO TIME T_P WITH SUBROUTINE ENCKE
II : N
<
3
2
≥
CALL WMATCN TO PREPARE [W] FOR PROPAGATION
$\left[W_{E,\ ICOM}\right] = [W]$
RETURN

Page 7 of 7

Figure B-58.- Concluded.


ENTER
$V_o^2 = V_o \cdot V_o$
$\alpha_i = r_i \Big/ \left(2 - r_i V_o^2 K_2^2\right)$
$\eta_i = \left(K_1^2 \big/ \alpha_i^3\right)^{\frac{1}{2}}$
$C_i = 1 - r_i \big/ \alpha_i$
$S_i = \left(\bar{r}_o \cdot \bar{V}_o\right) \Big/ \left(K_1^2 \alpha\right)^{\frac{1}{2}}$
EXIT

Page 1 of 1

Figure B-59. - Flow chart of subroutine OPS.

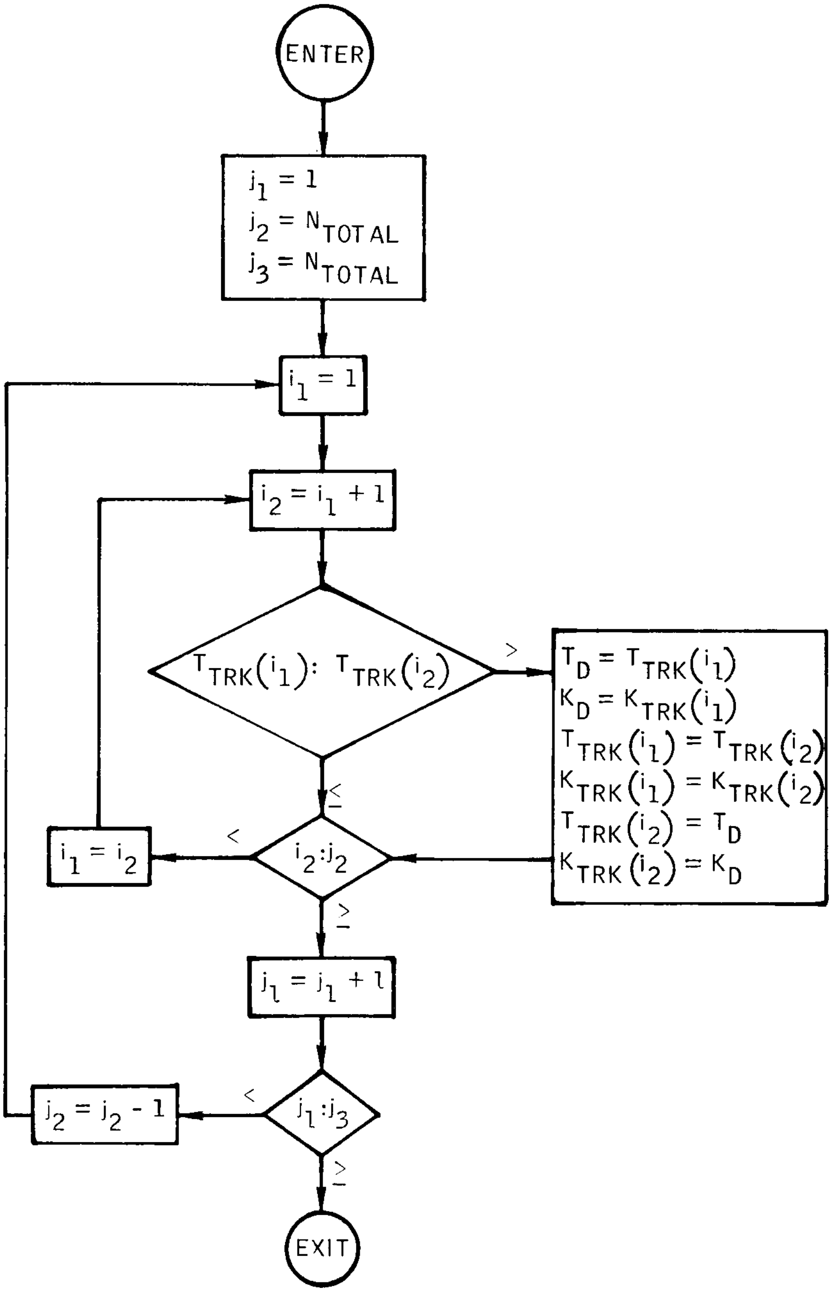
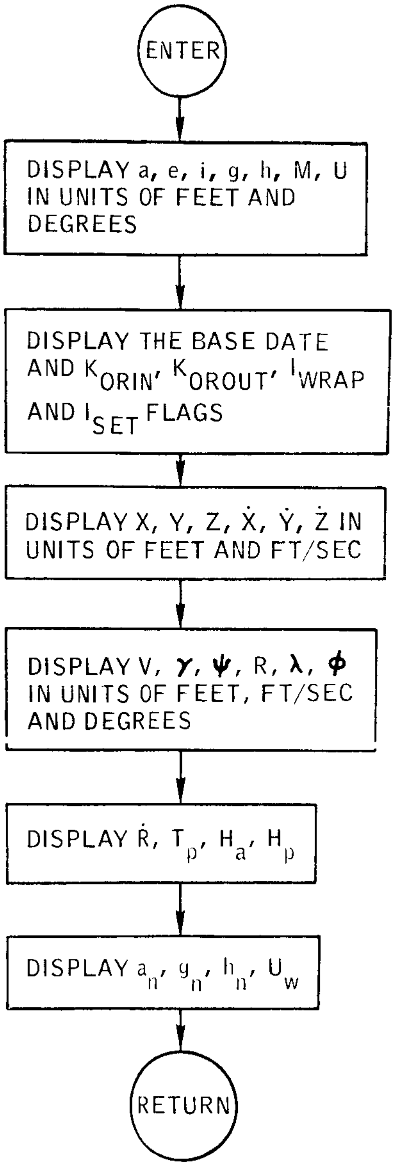


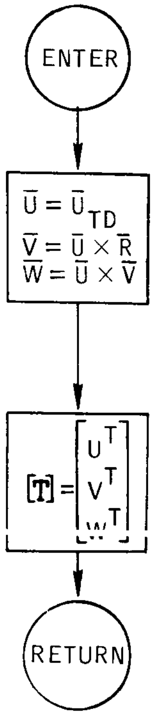
Figure B-60.- Flow chart of subroutine ORD.

Page 1 of 1


ENTER
DISPLAY a, e, i, g, h, M, U IN UNITS OF FEET AND DEGREES
DISPLAY THE BASE DATE AND K_{ORIN}, K_{OROUT}, I_{WRAP} AND I_{SET} FLAGS
DISPLAY X, Y, Z, $\dot{X}$, $\dot{Y}$, $\dot{Z}$ IN UNITS OF FEET AND FT/SEC
DISPLAY V, γ, ψ, R, λ, ϕ IN UNITS OF FEET, FT/SEC AND DEGREES
DISPLAY $\dot{R}$, T_p, H_a, H_p
DISPLAY a_n, g_n, h_n, U_w
RETURN

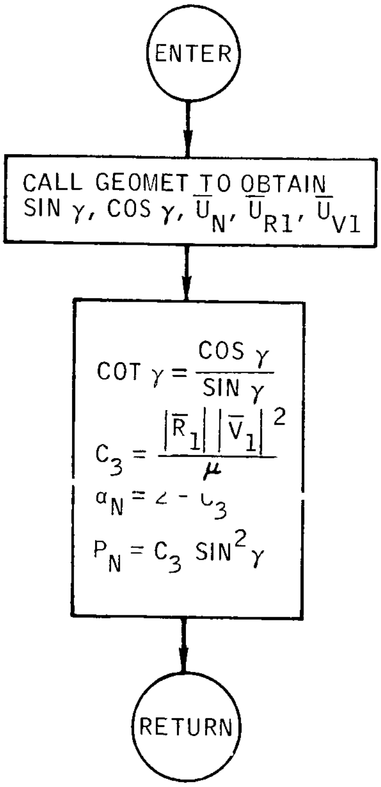
Page 1 of 1

Figure B-61.- Flow chart of subroutine OUTPUT.


ENTER
$\bar{U} = \bar{U}_{TD}$
$\bar{V} = \bar{U} \times \bar{R}$
$\bar{W} = \bar{U} \times \bar{V}$
$[T] = \begin{bmatrix} U^T \\ V^T \\ W^T \end{bmatrix}$
RETURN

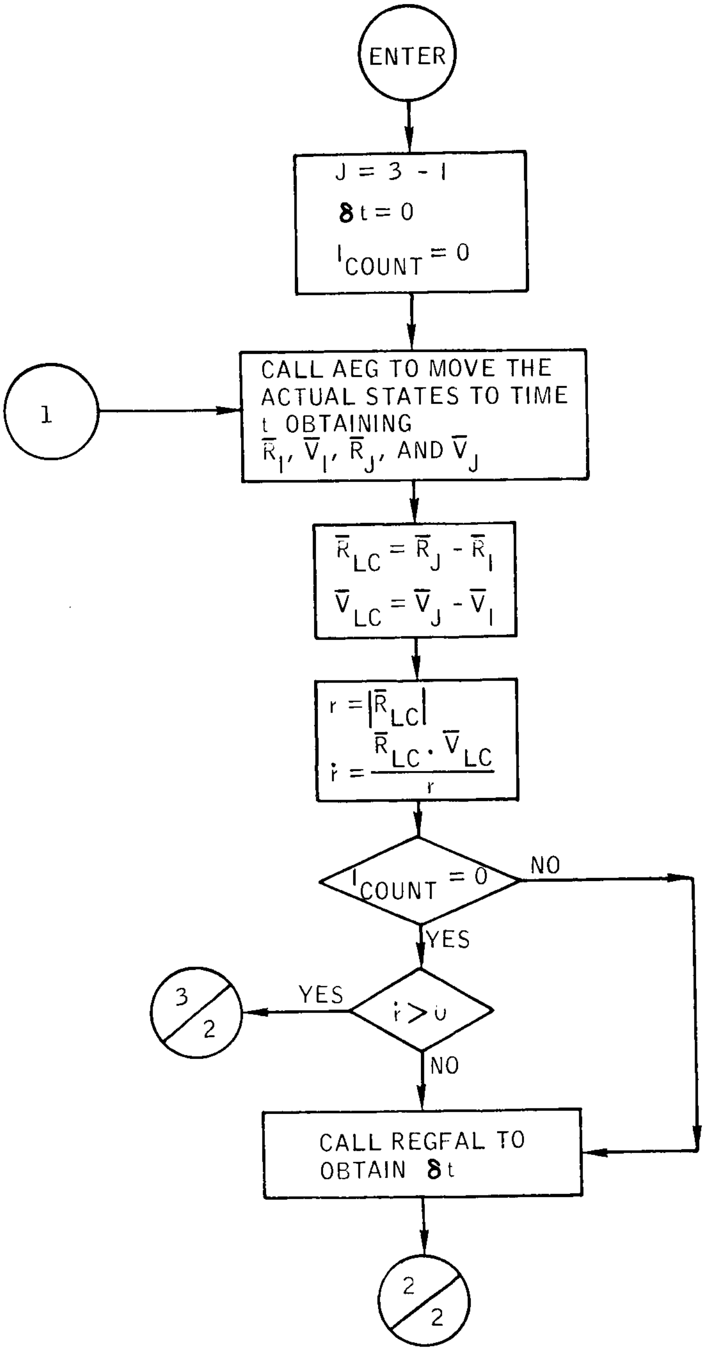
Page 1 of 1

Figure B-62.- Flow chart of subroutine PALIGN.


ENTER
CALL GEOMET TO OBTAIN SIN γ, COS γ, $\overline{U}_N$, $\overline{U}_{R1}$, $\overline{U}_{V1}$
COT $\gamma = \frac{\text{COS } \gamma}{\text{SIN } \gamma}$
$C_3 = \frac{\left|\overline{R}_1\right| \left|\overline{V}_1\right|^2}{\mu}$
$\alpha_N = 2 - C_3$
$P_N = C_3 \text{ SIN}^2 \gamma$
RETURN

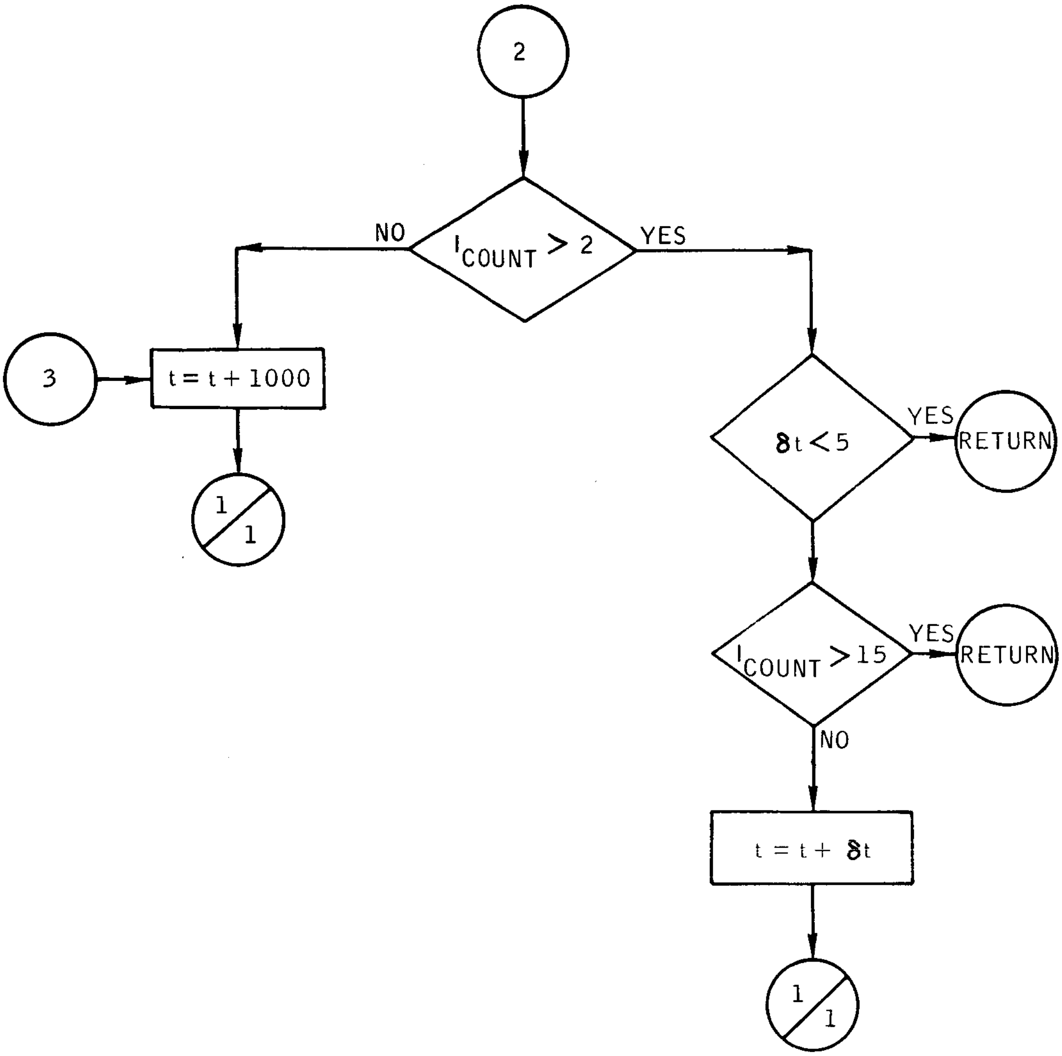
Page 1 of 1

Figure B-63.- Flow chart of subroutine PARAM.



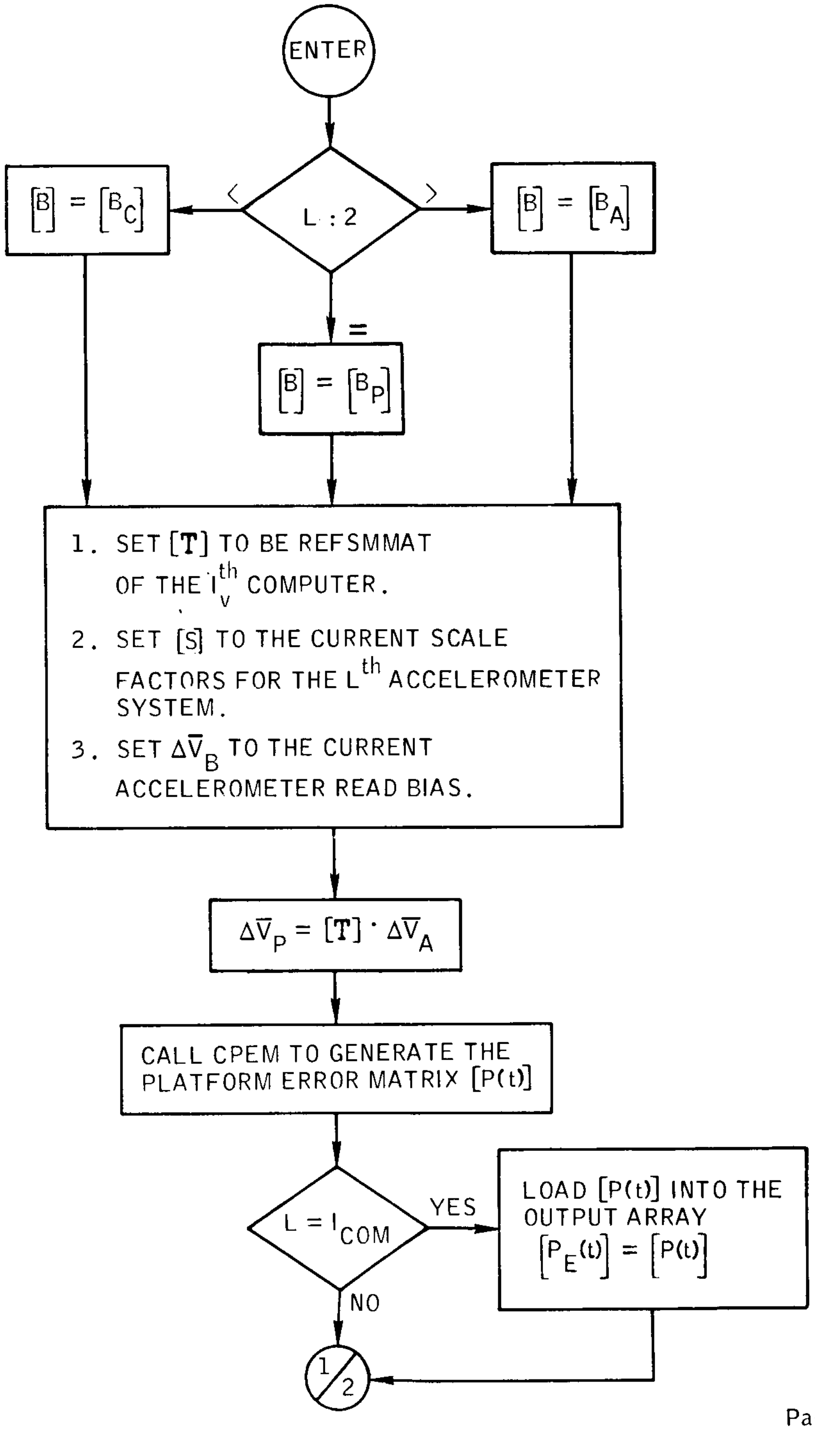
Page 1 of 2

Figure B-64.- Flow chart of subroutine PCA.



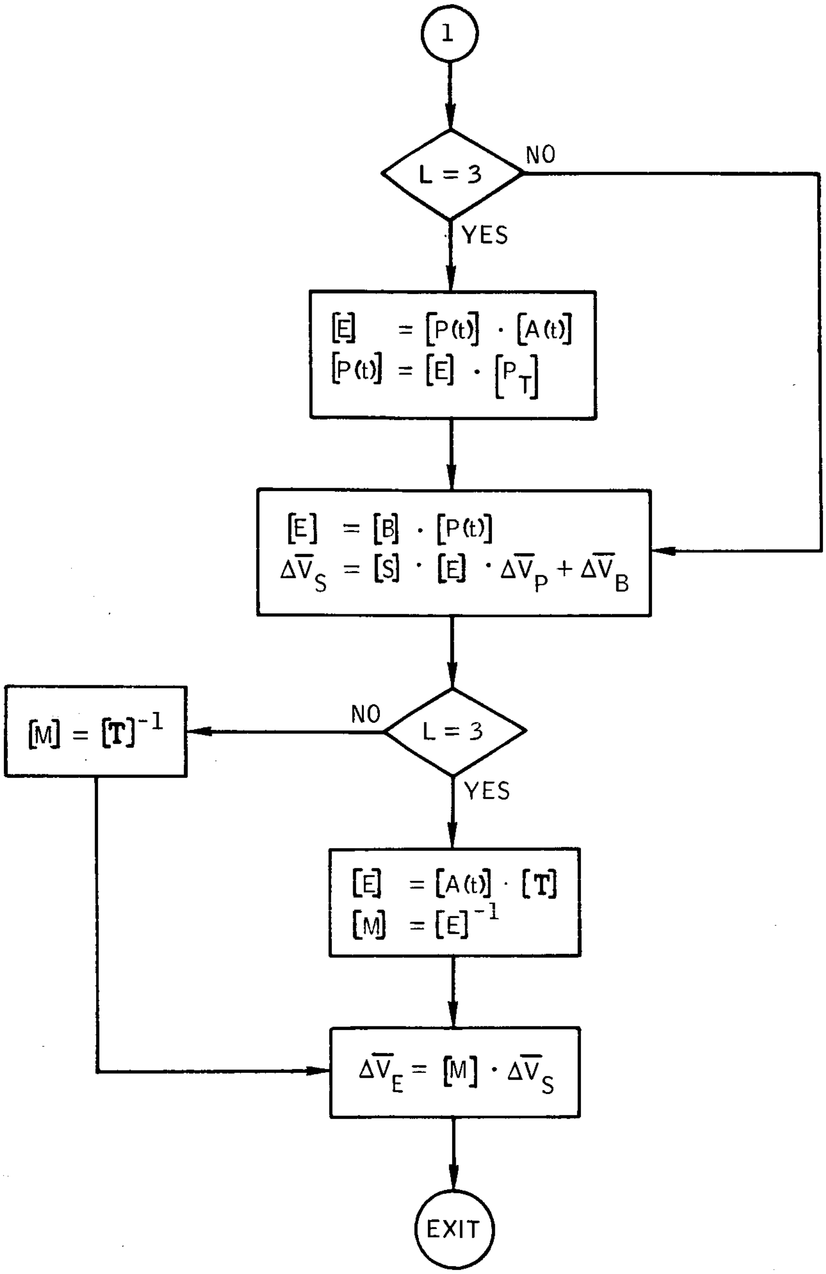
Page 2 of 2

Figure B-64.- Concluded.


ENTER
L : 2
<
>
=
[B] = [B_C]
[B] = [B_A]
[B] = [B_P]
1. SET [T] TO BE REFSMMAT OF THE I_v^{th} COMPUTER.
2. SET [S] TO THE CURRENT SCALE FACTORS FOR THE L^{th} ACCELEROMETER SYSTEM.
3. SET $\Delta\overline{V}_B$ TO THE CURRENT ACCELEROMETER READ BIAS.
$\Delta\overline{V}_P = [T] \cdot \Delta\overline{V}_A$
CALL CPEM TO GENERATE THE PLATFORM ERROR MATRIX [P(t)]
$L = I_{COM}$
YES
NO
LOAD [P(t)] INTO THE OUTPUT ARRAY
$[P_E(t)] = [P(t)]$
1/2

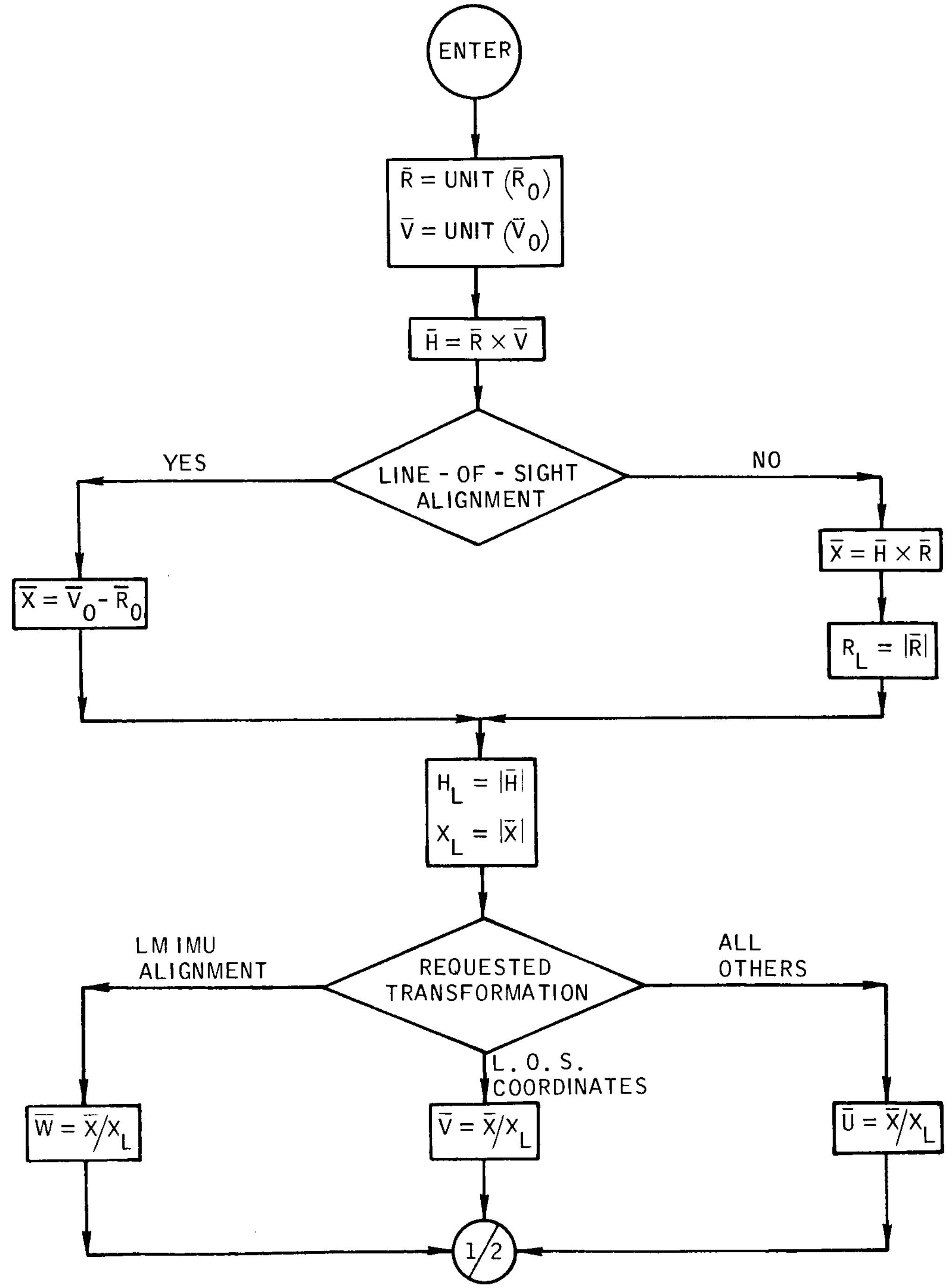
Page 1 of 2

Figure B-65.- Flow chart of subroutine PERRO.


1
L = 3
NO
YES
$[E] = [P(t)] \cdot [A(t)]$
$[P(t)] = [E] \cdot [P_T]$
$[E] = [B] \cdot [P(t)]$
$\Delta\overline{V}_S = [S] \cdot [E] \cdot \Delta\overline{V}_P + \Delta\overline{V}_B$
L = 3
NO
YES
$[M] = [\mathbf{T}]^{-1}$
$[E] = [A(t)] \cdot [\mathbf{T}]$
$[M] = [E]^{-1}$
$\Delta\overline{V}_E = [M] \cdot \Delta\overline{V}_S$
EXIT

Page 2 of 2

Figure B-65.- Concluded.



Page 1 of 3

Figure B-66.- Flow chart of subroutine PMATRX.

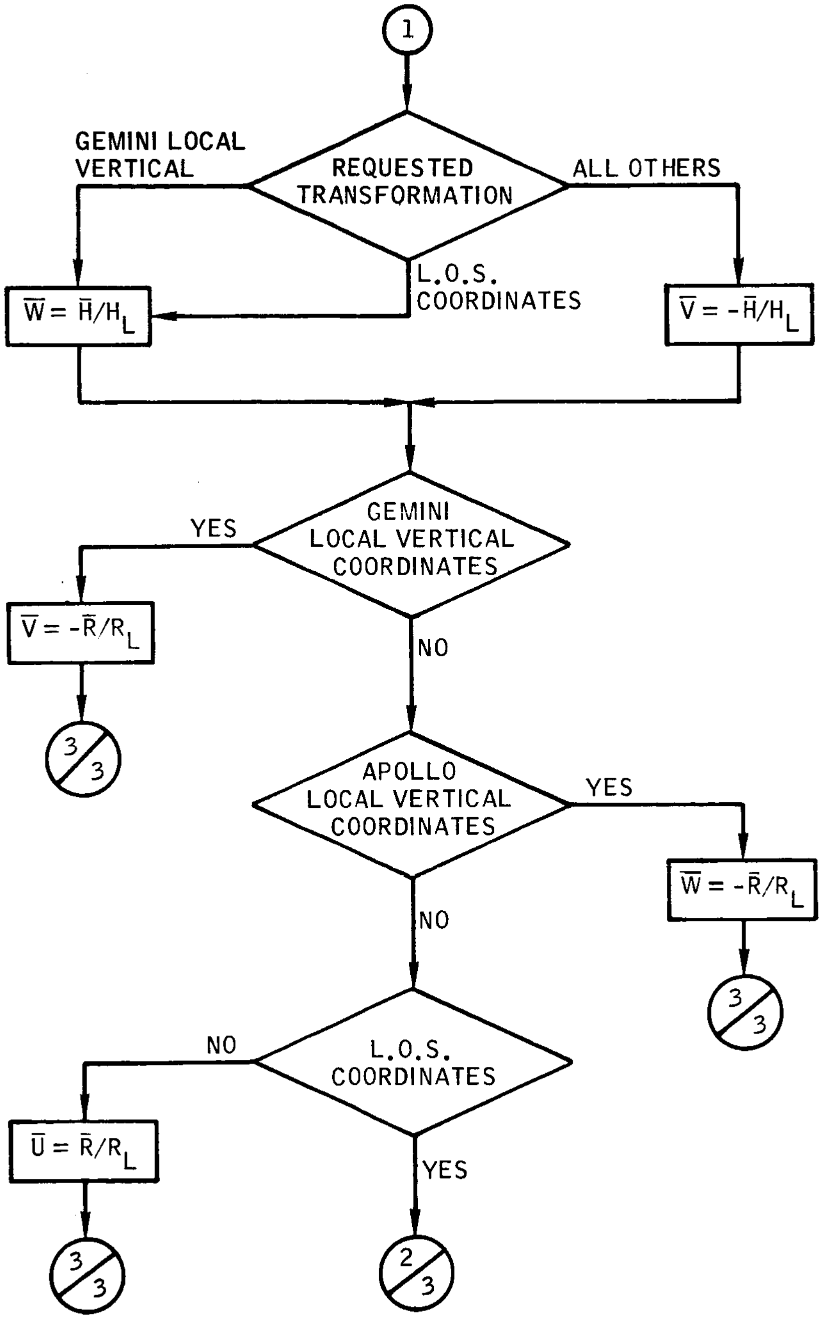
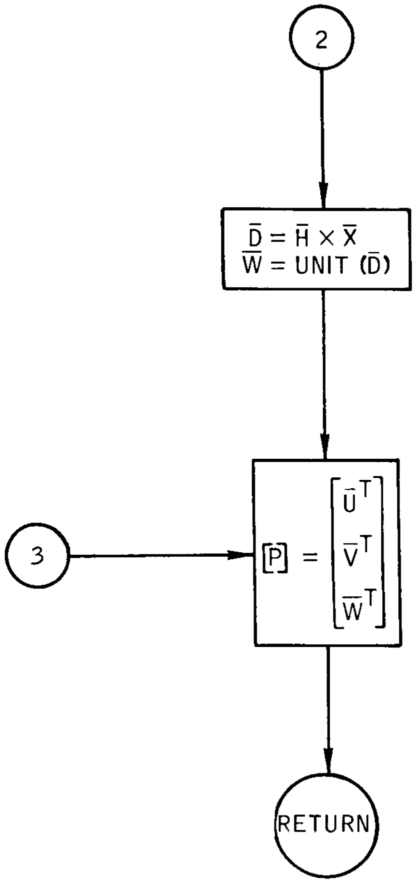

1
REQUESTED TRANSFORMATION
GEMINI LOCAL VERTICAL
ALL OTHERS
L.O.S. COORDINATES
$\overline{W} = \bar{H}/H_L$
$\overline{V} = -\bar{H}/H_L$
GEMINI LOCAL VERTICAL COORDINATES
YES
NO
$\overline{V} = -\bar{R}/R_L$
3/3
APOLLO LOCAL VERTICAL COORDINATES
YES
NO
$\overline{W} = -\bar{R}/R_L$
3/3
L.O.S. COORDINATES
NO
YES
$\overline{U} = \bar{R}/R_L$
3/3
2/3

Figure B-66.- Continued.

Page 2 of 3


2
$\bar{D} = \bar{H} \times \bar{X}$
$\bar{W} = \text{UNIT } (\bar{D})$
3
$[P] = \begin{bmatrix} \bar{U}^T \\ \bar{V}^T \\ \bar{W}^T \end{bmatrix}$
RETURN

Page 3 of 3

Figure B-66.- Concluded.

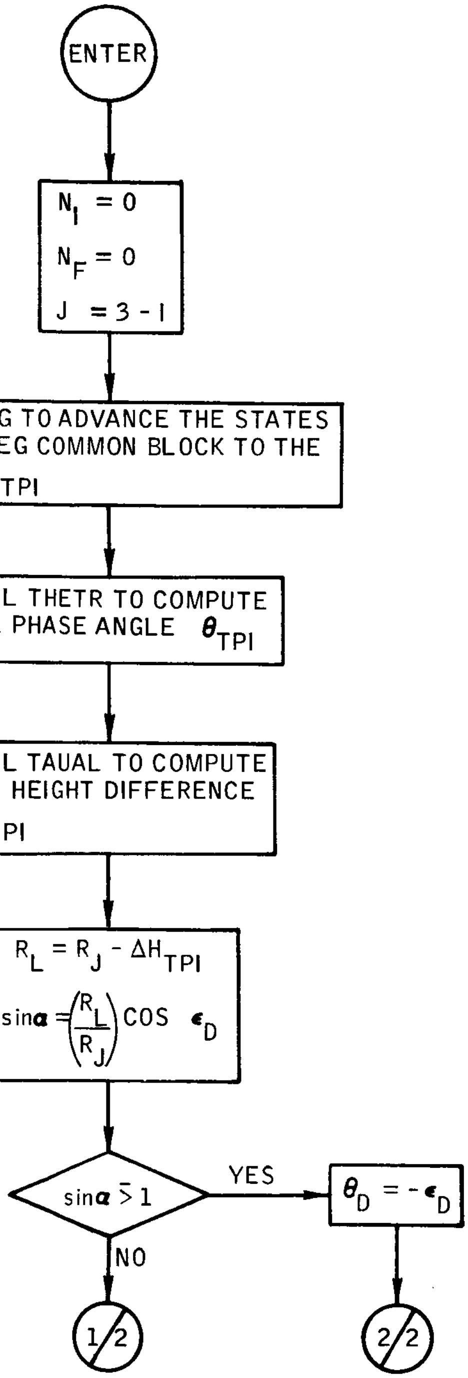

ENTER
$N_I = 0$
$N_F = 0$
$J = 3 - I$
CALL AEG TO ADVANCE THE STATES IN THE AEG COMMON BLOCK TO THE TIME T_{TPI}
CALL THETR TO COMPUTE THE PHASE ANGLE θ_{TPI}
CALL TAUAL TO COMPUTE THE HEIGHT DIFFERENCE ΔH_{TPI}
$R_L = R_J - \Delta H_{TPI}$
$\sin\alpha = \left(\frac{R_L}{R_J}\right) \cos \epsilon_D$
$\sin\alpha \geq 1$
YES
$\theta_D = -\epsilon_D$
NO
1/2
2/2

Figure B-67.- Flow chart of subroutine PMISS.

Page 1 of 2

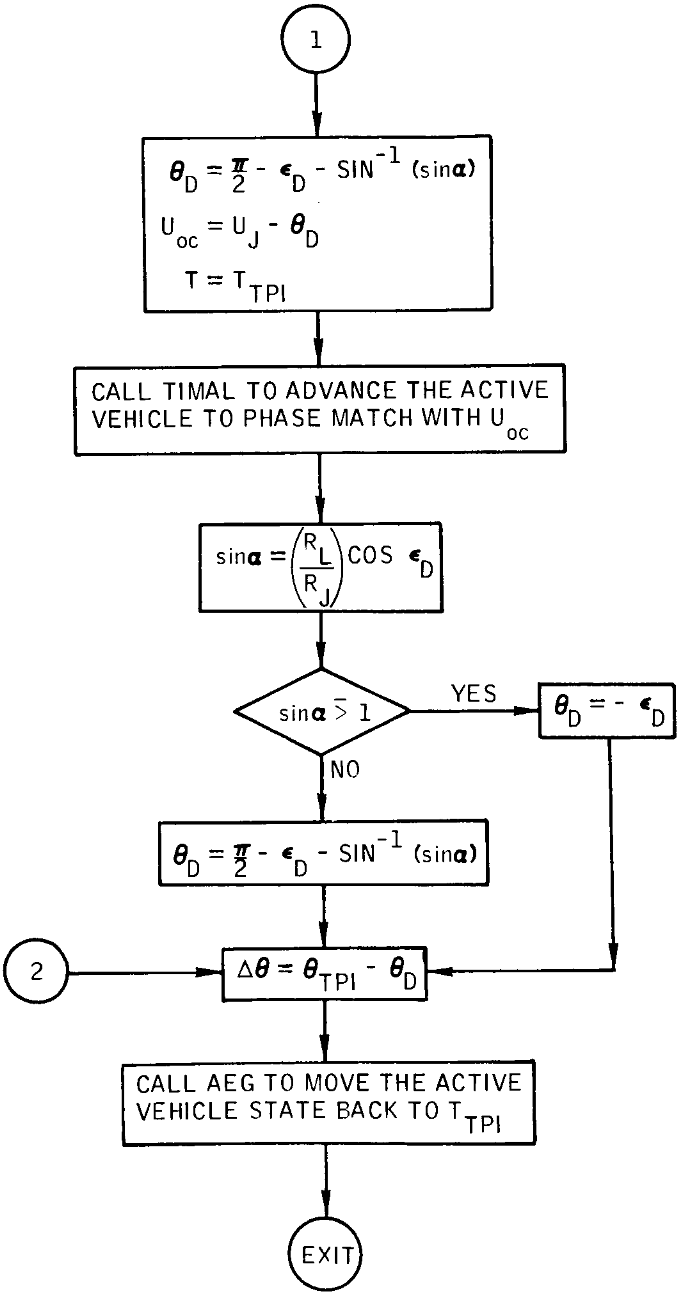

1
$\theta_D = \frac{\pi}{2} - \epsilon_D - \mathrm{SIN}^{-1} (\sin\alpha)$
$U_{oc} = U_J - \theta_D$
$T = T_{TPI}$
CALL TIMAL TO ADVANCE THE ACTIVE VEHICLE TO PHASE MATCH WITH U_{oc}
$\sin\alpha = \left(\frac{R_L}{R_J}\right) \mathrm{COS}\ \epsilon_D$
$\sin\alpha \overset{=}{>} 1$
YES
$\theta_D = -\epsilon_D$
NO
$\theta_D = \frac{\pi}{2} - \epsilon_D - \mathrm{SIN}^{-1} (\sin\alpha)$
2
$\Delta\theta = \theta_{TPI} - \theta_D$
CALL AEG TO MOVE THE ACTIVE VEHICLE STATE BACK TO T_{TPI}
EXIT

Figure B-67.- Concluded.

Page 2 of 2

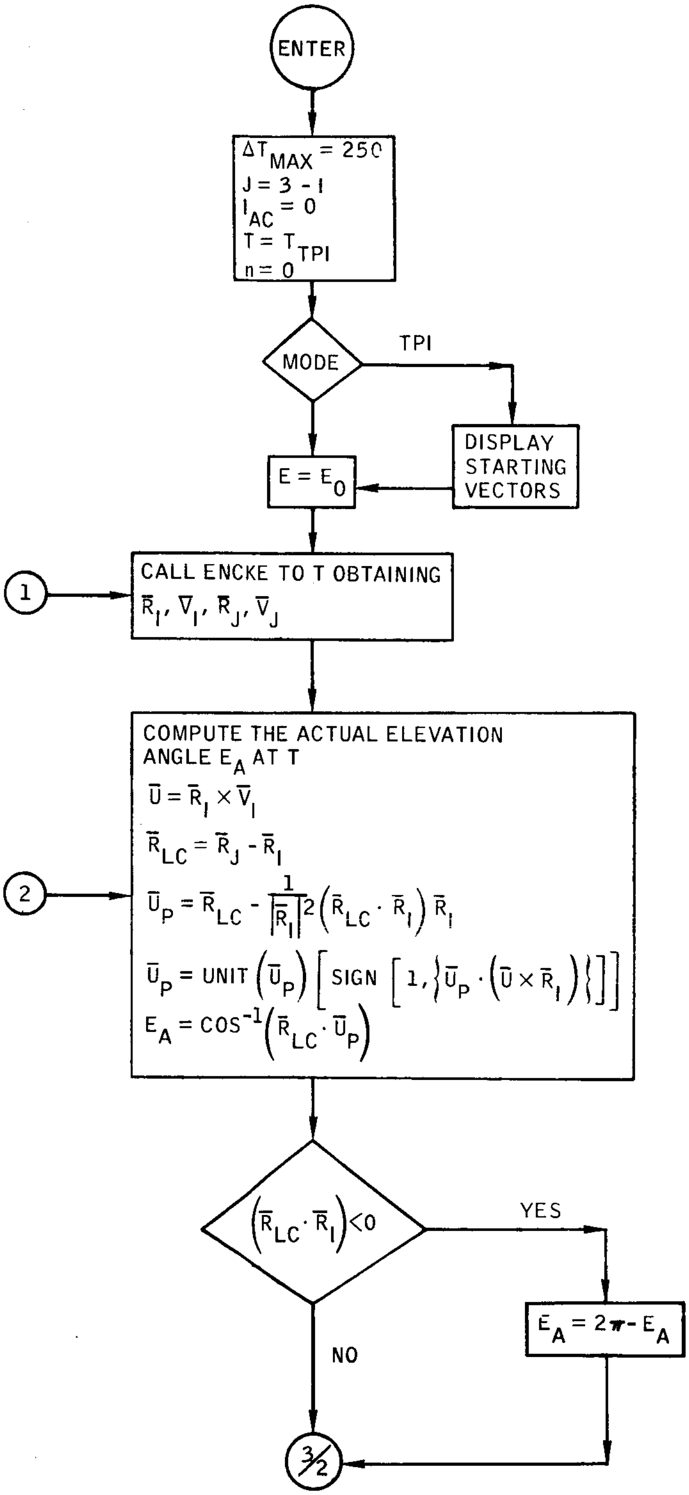
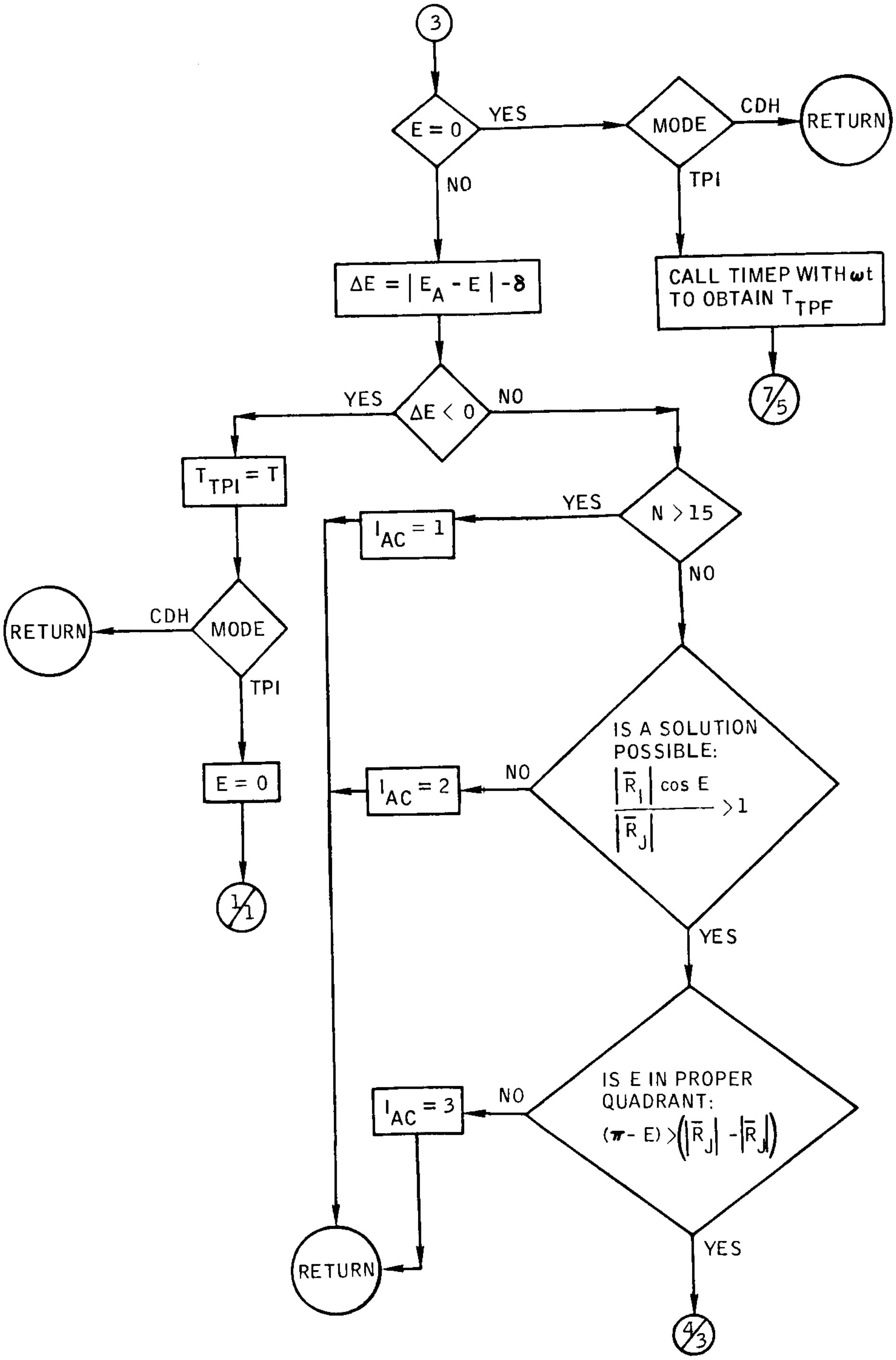

ENTER
$\Delta T_{MAX} = 250$
$J = 3 - I$
$I_{AC} = 0$
$T = T_{TPI}$
$n = 0$
MODE
TPI
DISPLAY STARTING VECTORS
$E = E_0$
1
CALL ENCKE TO T OBTAINING
$\bar{R}_I, \bar{V}_I, \bar{R}_J, \bar{V}_J$
COMPUTE THE ACTUAL ELEVATION ANGLE E_A AT T
$\bar{U} = \bar{R}_I \times \bar{V}_I$
$\bar{R}_{LC} = \bar{R}_J - \bar{R}_I$
2
$\bar{U}_P = \bar{R}_{LC} - \frac{1}{|\bar{R}_I|^2}\left(\bar{R}_{LC} \cdot \bar{R}_I\right)\bar{R}_I$
$\bar{U}_P = \text{UNIT}\left(\bar{U}_P\right)\left[\text{SIGN}\left[1,\left\}\bar{U}_P \cdot \left(\bar{U} \times \bar{R}_I\right)\right\{\right]\right]$
$E_A = \cos^{-1}\left(\bar{R}_{LC} \cdot \bar{U}_P\right)$
$\left(\bar{R}_{LC} \cdot \bar{R}_I\right) < 0$
YES
$E_A = 2\pi - E_A$
NO
3/2

Figure B-68.- Flow chart of subroutine PRETPI.

Page 1 of 5



Page 2 of 5

Figure B-68.- Continued.

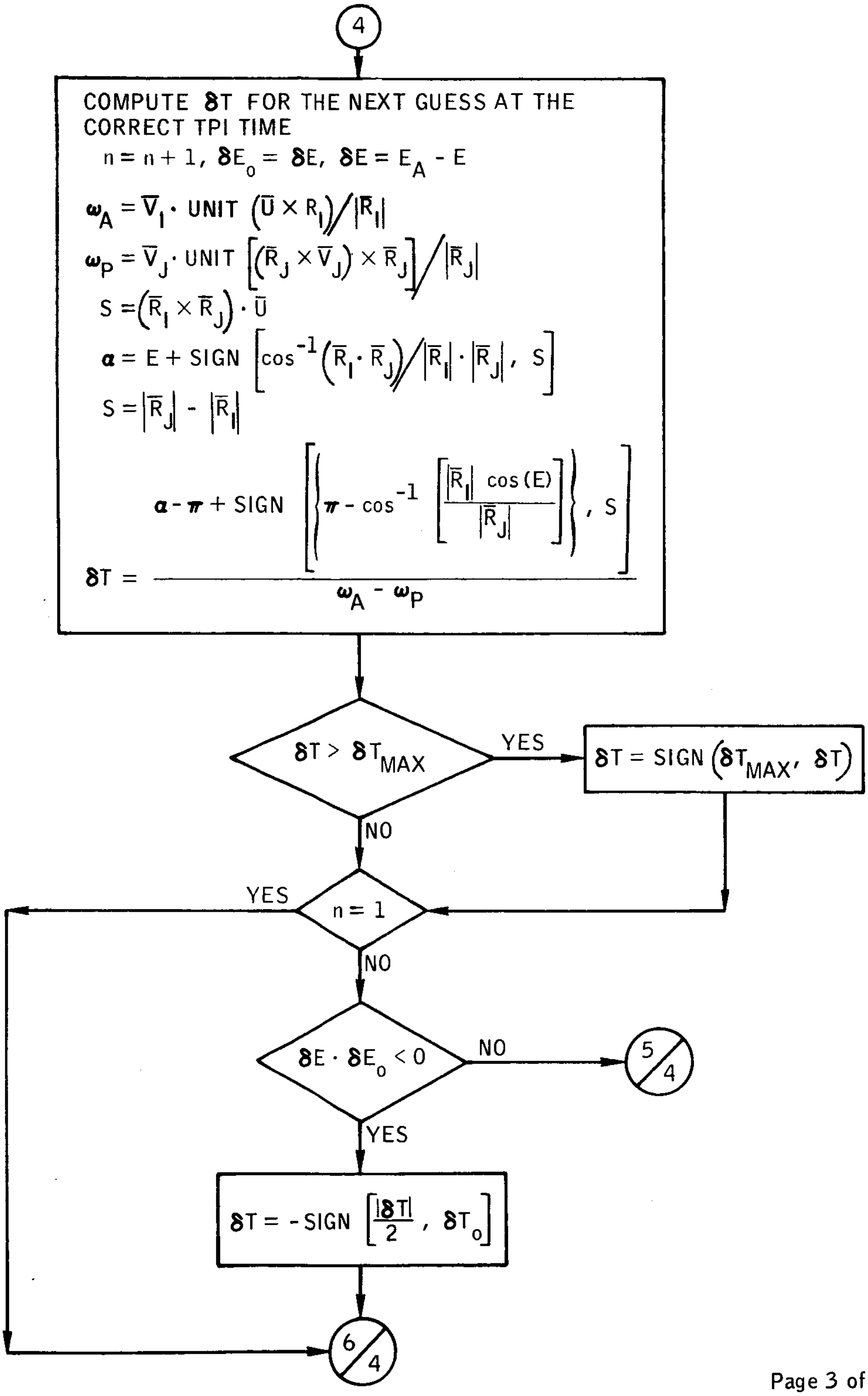
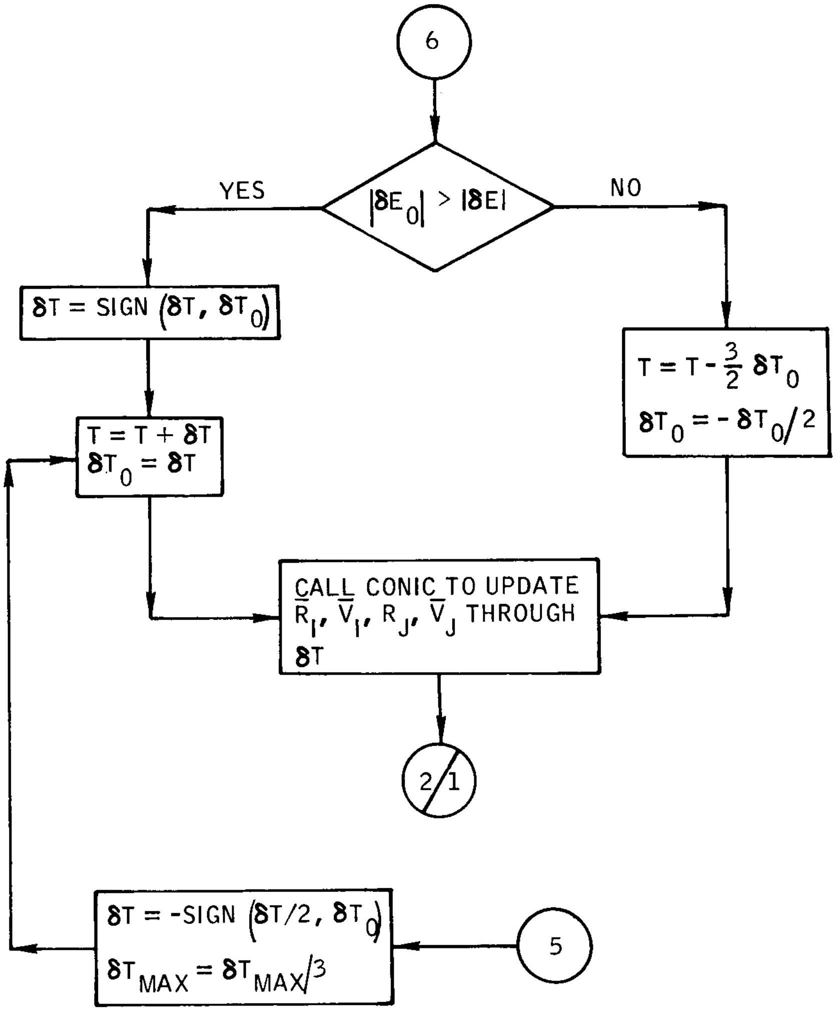
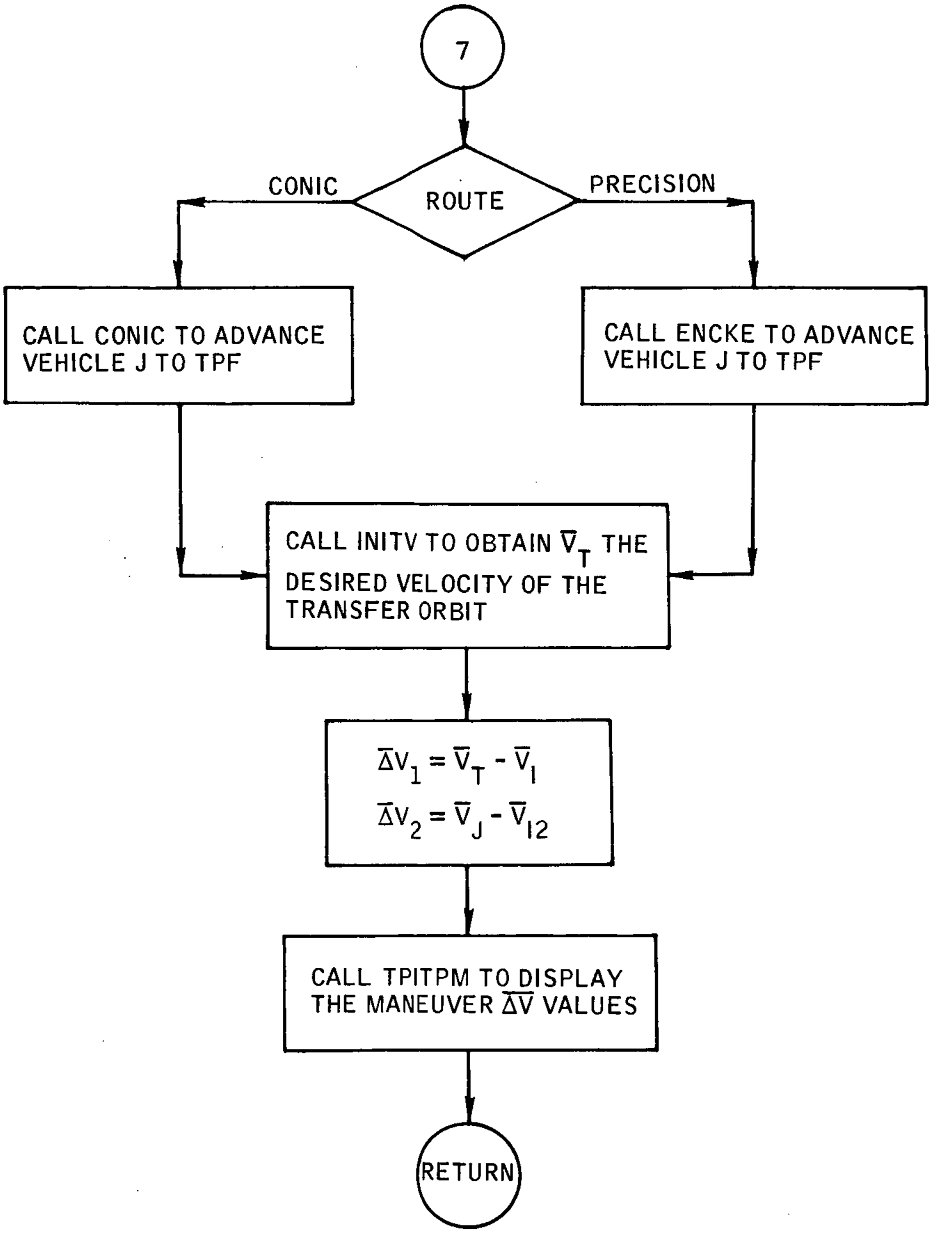

4
COMPUTE δT FOR THE NEXT GUESS AT THE CORRECT TPI TIME
$n = n+1,\ \delta E_0 = \delta E,\ \delta E = E_A - E$
$\omega_A = \bar{V}_I \cdot \text{UNIT}\left(\bar{U} \times R_I\right) / |\bar{R}_I|$
$\omega_P = \bar{V}_J \cdot \text{UNIT}\left[\left(\bar{R}_J \times \bar{V}_J\right) \times \bar{R}_J\right] / |\bar{R}_J|$
$S = \left(\bar{R}_I \times \bar{R}_J\right) \cdot \bar{U}$
$\alpha = E + \text{SIGN}\left[\cos^{-1}\left(\bar{R}_I \cdot \bar{R}_J\right) / |\bar{R}_I| \cdot |\bar{R}_J|,\ S\right]$
$S = |\bar{R}_J| - |\bar{R}_I|$
$\delta T = \dfrac{\alpha - \pi + \text{SIGN}\left[\left\{\pi - \cos^{-1}\left[\dfrac{|\bar{R}_I| \cos(E)}{|\bar{R}_J|}\right]\right\},\ S\right]}{\omega_A - \omega_P}$
$\delta T > \delta T_{MAX}$
YES
$\delta T = \text{SIGN}\left(\delta T_{MAX},\ \delta T\right)$
NO
$n = 1$
YES
NO
$\delta E \cdot \delta E_0 < 0$
NO
5/4
YES
$\delta T = -\text{SIGN}\left[\dfrac{|\delta T|}{2},\ \delta T_0\right]$
6/4

Figure B-68.- Continued.


6
$|\delta E_0| > |\delta E|$
YES
NO
$\delta T = SIGN(\delta T, \delta T_0)$
$T = T - \frac{3}{2}\delta T_0$
$\delta T_0 = -\delta T_0/2$
$T = T + \delta T$
$\delta T_0 = \delta T$
CALL CONIC TO UPDATE $\overline{R}_I$, $\overline{V}_I$, R_J, $\overline{V}_J$ THROUGH δT
2/1
$\delta T = -SIGN(\delta T/2, \delta T_0)$
$\delta T_{MAX} = \delta T_{MAX}/3$
5

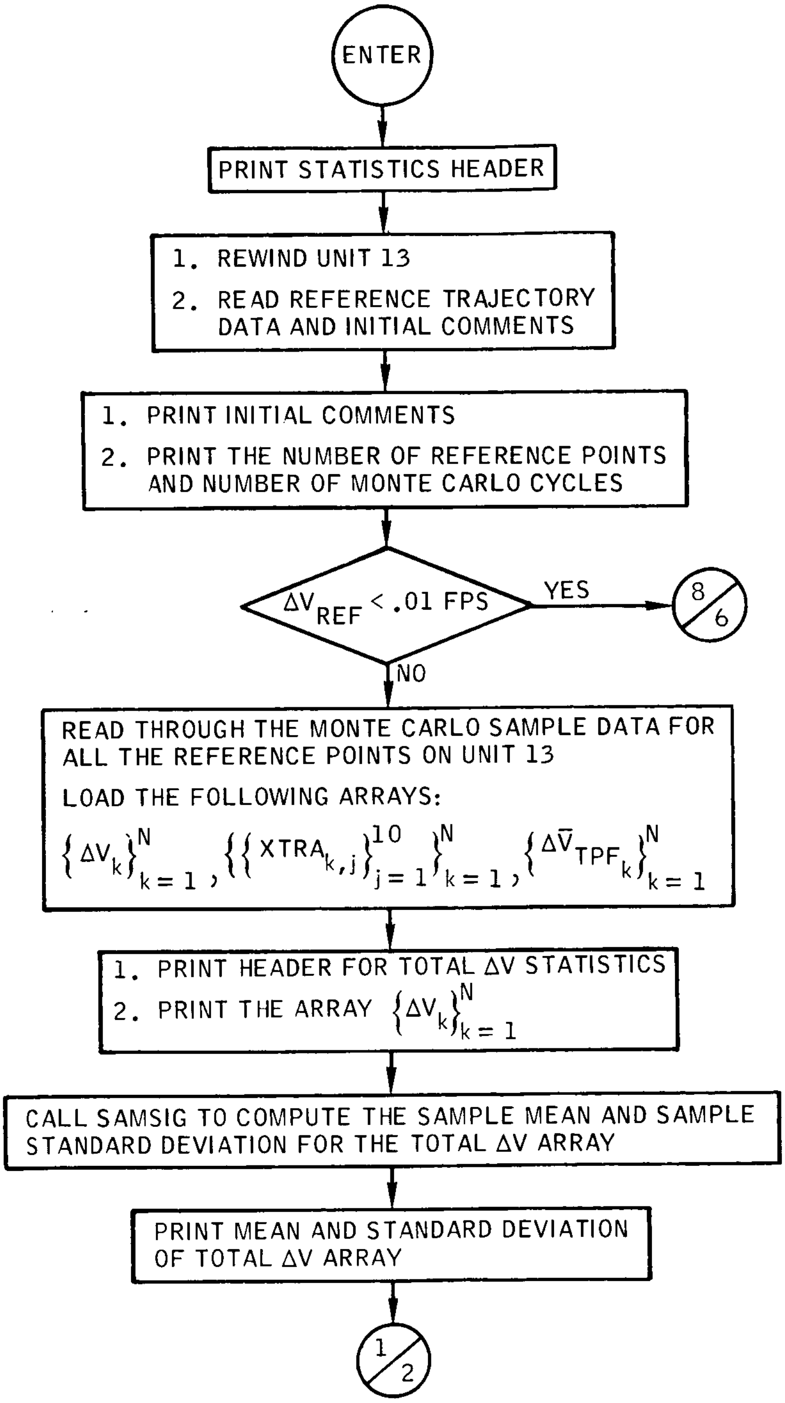
Page 4 of 5

Figure B-68.- Continued.


7
ROUTE
CONIC
PRECISION
CALL CONIC TO ADVANCE VEHICLE J TO TPF
CALL ENCKE TO ADVANCE VEHICLE J TO TPF
CALL INITV TO OBTAIN $\overline{V}_T$ THE DESIRED VELOCITY OF THE TRANSFER ORBIT
$\overline{\Delta}V_1 = \overline{V}_T - \overline{V}_I$
$\overline{\Delta}V_2 = \overline{V}_J - \overline{V}_{I2}$
CALL TPITPM TO DISPLAY THE MANEUVER $\overline{\Delta V}$ VALUES
RETURN

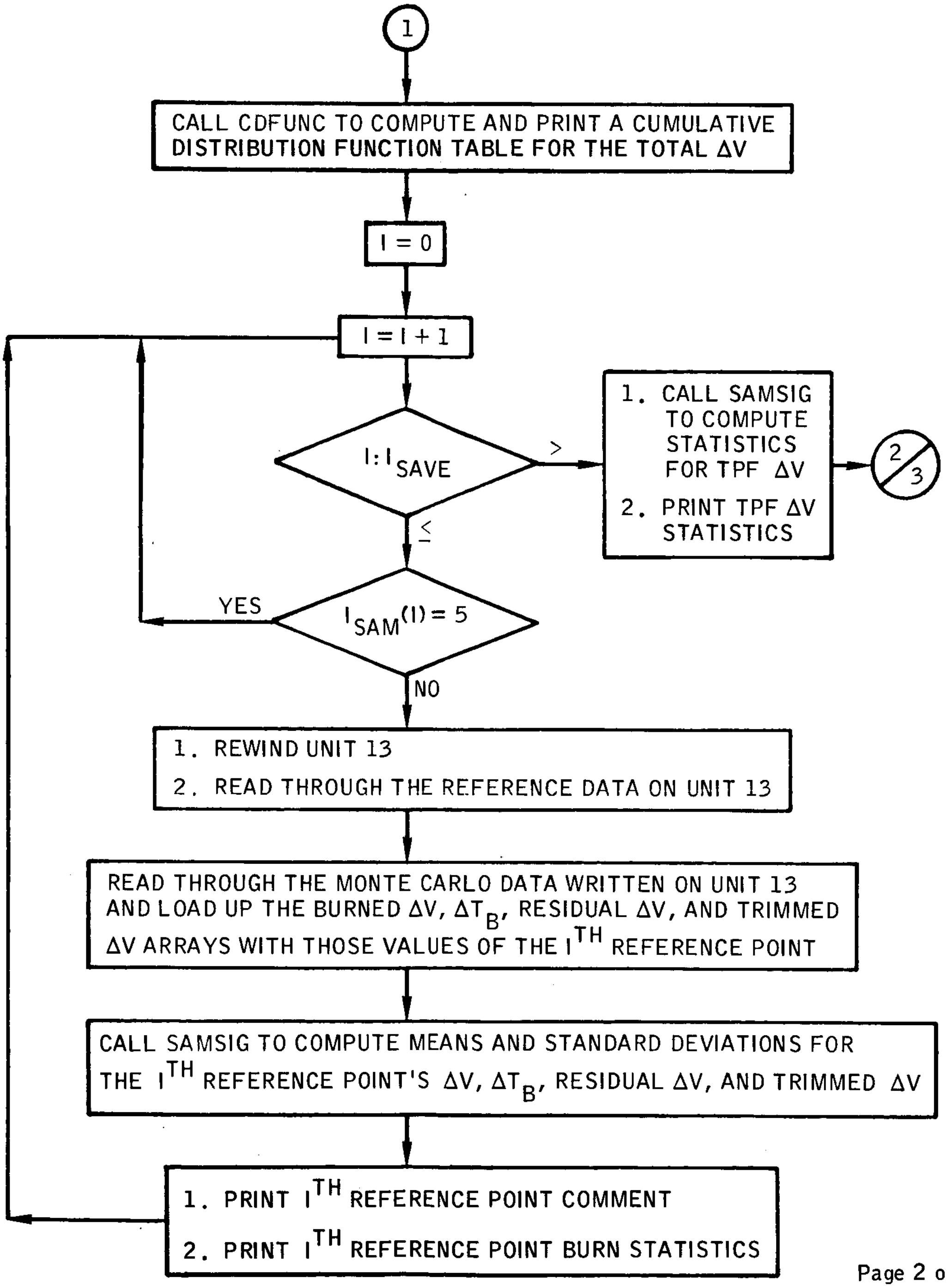
Page 5 of 5

Figure B-68.- Concluded.


ENTER
PRINT STATISTICS HEADER
1. REWIND UNIT 13
2. READ REFERENCE TRAJECTORY DATA AND INITIAL COMMENTS
1. PRINT INITIAL COMMENTS
2. PRINT THE NUMBER OF REFERENCE POINTS AND NUMBER OF MONTE CARLO CYCLES
$\Delta V_{REF} < .01$ FPS
YES
8/6
NO
READ THROUGH THE MONTE CARLO SAMPLE DATA FOR ALL THE REFERENCE POINTS ON UNIT 13
LOAD THE FOLLOWING ARRAYS:
$\left\{\Delta V_k\right\}_{k=1}^{N}, \left\{\left\{XTRA_{k,j}\right\}_{j=1}^{10}\right\}_{k=1}^{N}, \left\{\Delta \bar{V}_{TPF_k}\right\}_{k=1}^{N}$
1. PRINT HEADER FOR TOTAL ΔV STATISTICS
2. PRINT THE ARRAY $\left\{\Delta V_k\right\}_{k=1}^{N}$
CALL SAMSIG TO COMPUTE THE SAMPLE MEAN AND SAMPLE STANDARD DEVIATION FOR THE TOTAL ΔV ARRAY
PRINT MEAN AND STANDARD DEVIATION OF TOTAL ΔV ARRAY
1/2

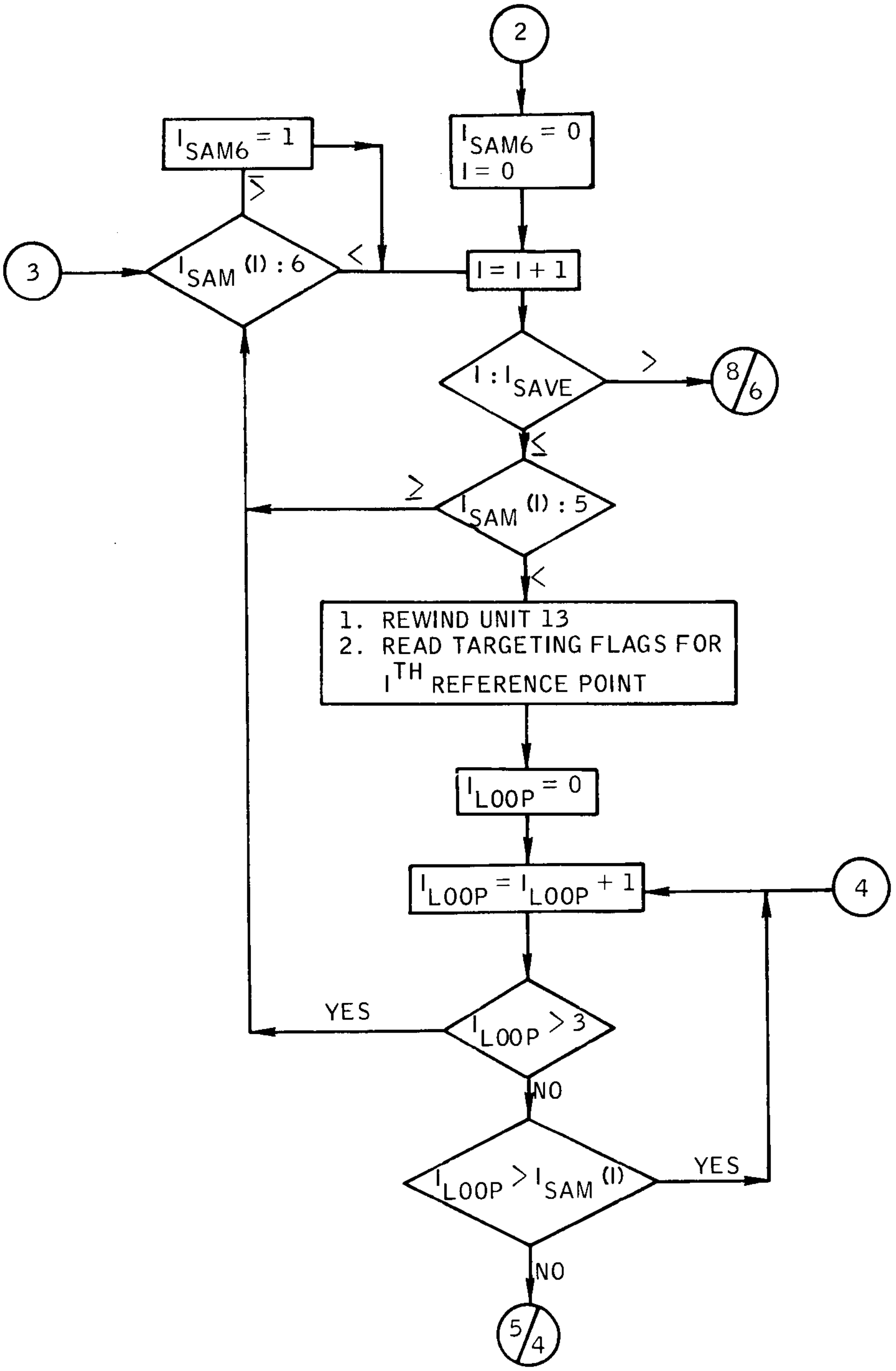
Page 1 of 7

Figure B-69.- Flow chart of subroutine PROCES.


1
CALL CDFUNC TO COMPUTE AND PRINT A CUMULATIVE DISTRIBUTION FUNCTION TABLE FOR THE TOTAL ΔV
I = 0
I = I + 1
I : I_SAVE
>
1. CALL SAMSIG TO COMPUTE STATISTICS FOR TPF ΔV
2. PRINT TPF ΔV STATISTICS
2/3
≤
I_SAM(I) = 5
YES
NO
1. REWIND UNIT 13
2. READ THROUGH THE REFERENCE DATA ON UNIT 13
READ THROUGH THE MONTE CARLO DATA WRITTEN ON UNIT 13 AND LOAD UP THE BURNED ΔV, ΔT_B, RESIDUAL ΔV, AND TRIMMED ΔV ARRAYS WITH THOSE VALUES OF THE I^TH REFERENCE POINT
CALL SAMSIG TO COMPUTE MEANS AND STANDARD DEVIATIONS FOR THE I^TH REFERENCE POINT'S ΔV, ΔT_B, RESIDUAL ΔV, AND TRIMMED ΔV
1. PRINT I^TH REFERENCE POINT COMMENT
2. PRINT I^TH REFERENCE POINT BURN STATISTICS

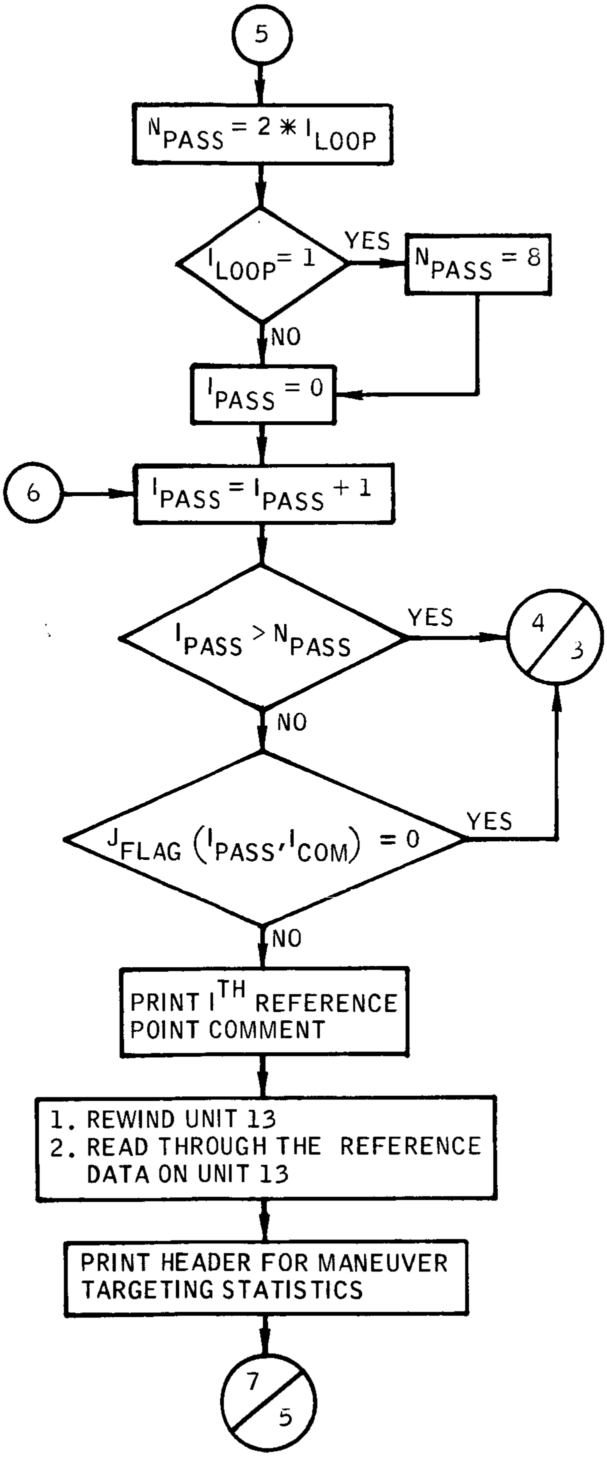
Page 2 of 7

Figure B-69.- Continued.



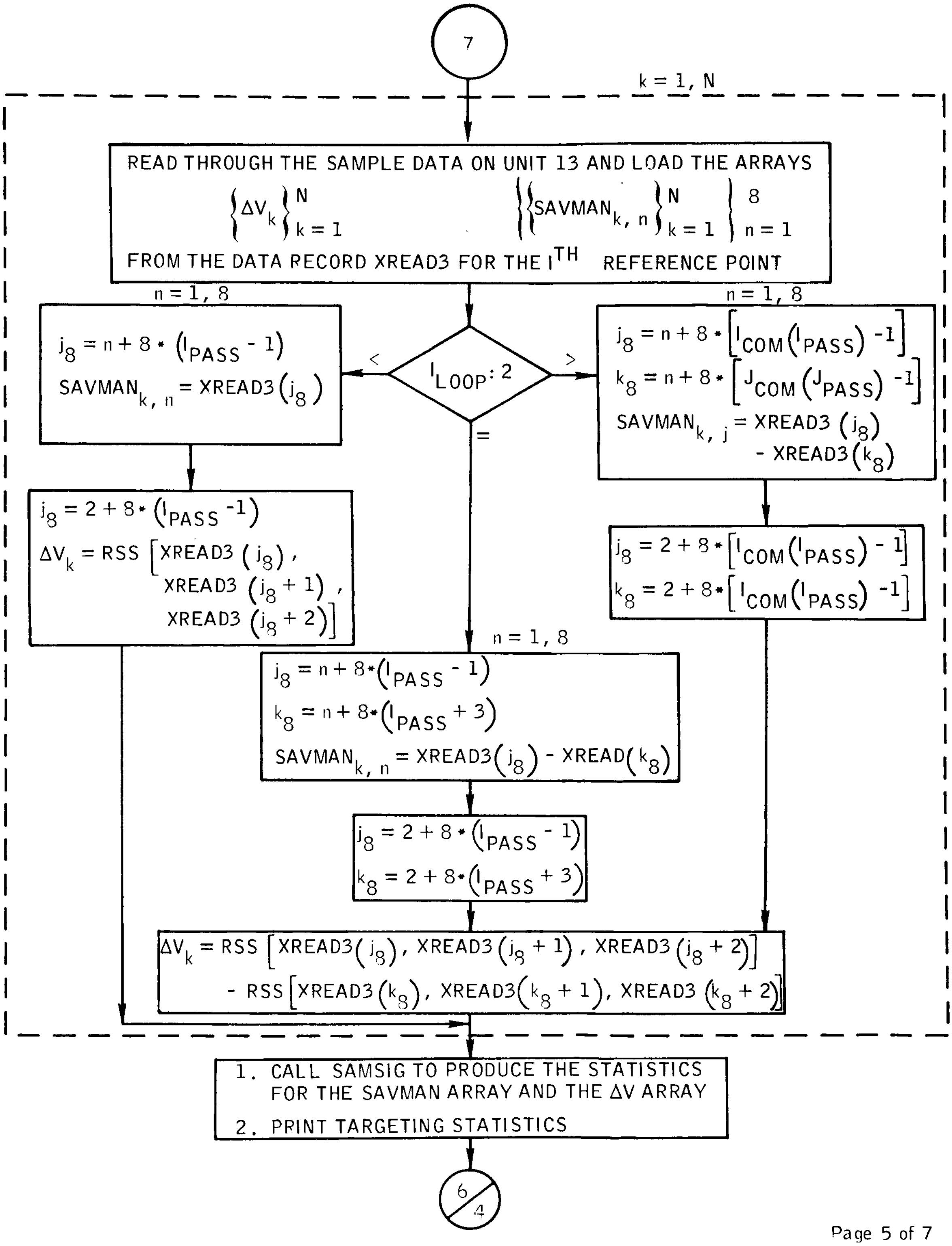
Page 3 of 7

Figure B-69.- Continued.


5
$N_{PASS} = 2 * I_{LOOP}$
$I_{LOOP} = 1$
YES
$N_{PASS} = 8$
NO
$I_{PASS} = 0$
6
$I_{PASS} = I_{PASS} + 1$
$I_{PASS} > N_{PASS}$
YES
4/3
NO
$J_{FLAG} (I_{PASS}, I_{COM}) = 0$
YES
NO
PRINT I^{TH} REFERENCE POINT COMMENT
1. REWIND UNIT 13
2. READ THROUGH THE REFERENCE DATA ON UNIT 13
PRINT HEADER FOR MANEUVER TARGETING STATISTICS
7/5

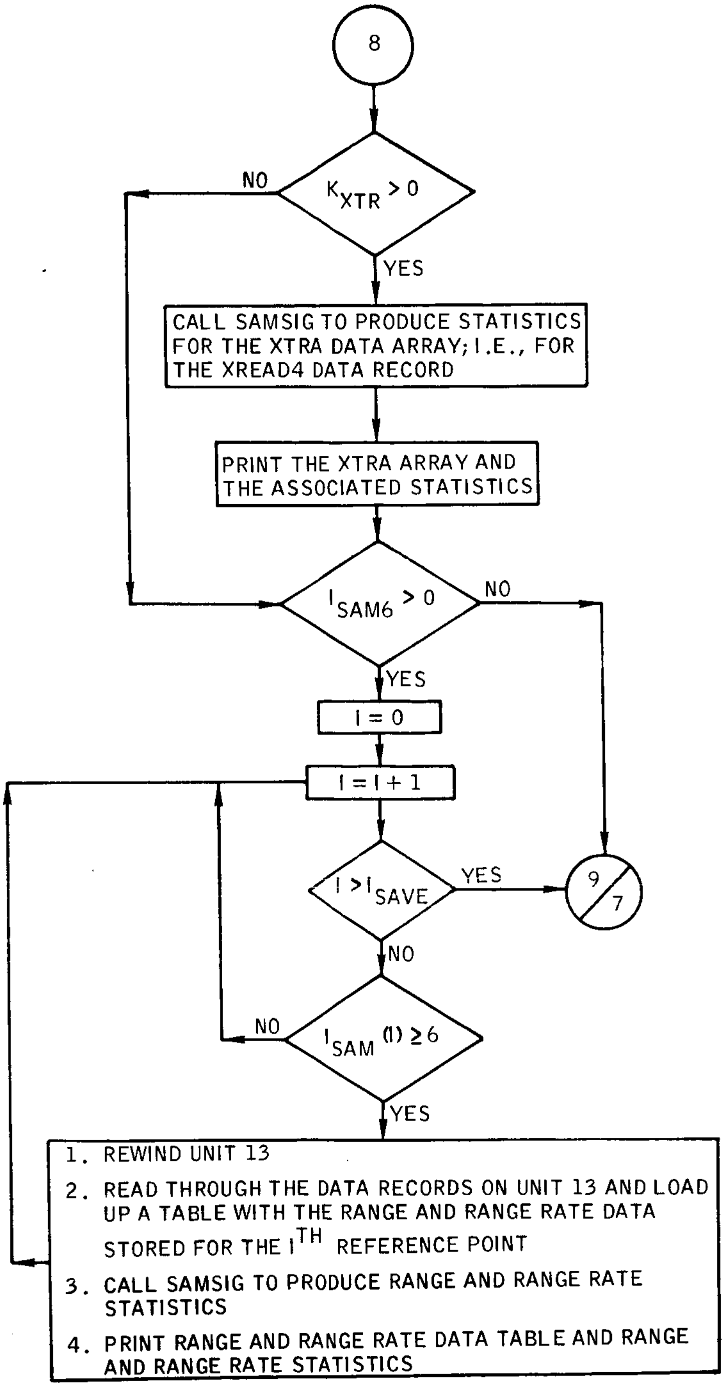
Page 4 of 7

Figure B-69.- Continued.


7
k = 1, N
READ THROUGH THE SAMPLE DATA ON UNIT 13 AND LOAD THE ARRAYS
$\{\Delta V_k\}_{k=1}^{N}$
$\{\{SAVMAN_{k,n}\}_{k=1}^{N}\}_{n=1}^{8}$
FROM THE DATA RECORD XREAD3 FOR THE I^{TH} REFERENCE POINT
n = 1, 8
$j_8 = n + 8 * (I_{PASS} - 1)$
$SAVMAN_{k,n} = XREAD3(j_8)$
<
$I_{LOOP}: 2$
>
=
n = 1, 8
$j_8 = n + 8 * [I_{COM}(I_{PASS}) - 1]$
$k_8 = n + 8 * [J_{COM}(J_{PASS}) - 1]$
$SAVMAN_{k,j} = XREAD3(j_8) - XREAD3(k_8)$
$j_8 = 2 + 8 * (I_{PASS} - 1)$
$\Delta V_k = RSS[XREAD3(j_8), XREAD3(j_8 + 1), XREAD3(j_8 + 2)]$
$j_8 = 2 + 8 * [I_{COM}(I_{PASS}) - 1]$
$k_8 = 2 + 8 * [I_{COM}(I_{PASS}) - 1]$
n = 1, 8
$j_8 = n + 8 * (I_{PASS} - 1)$
$k_8 = n + 8 * (I_{PASS} + 3)$
$SAVMAN_{k,n} = XREAD3(j_8) - XREAD(k_8)$
$j_8 = 2 + 8 * (I_{PASS} - 1)$
$k_8 = 2 + 8 * (I_{PASS} + 3)$
$\Delta V_k = RSS[XREAD3(j_8), XREAD3(j_8 + 1), XREAD3(j_8 + 2)] - RSS[XREAD3(k_8), XREAD3(k_8 + 1), XREAD3(k_8 + 2)]$
1. CALL SAMSIG TO PRODUCE THE STATISTICS FOR THE SAVMAN ARRAY AND THE ΔV ARRAY
2. PRINT TARGETING STATISTICS
6/4

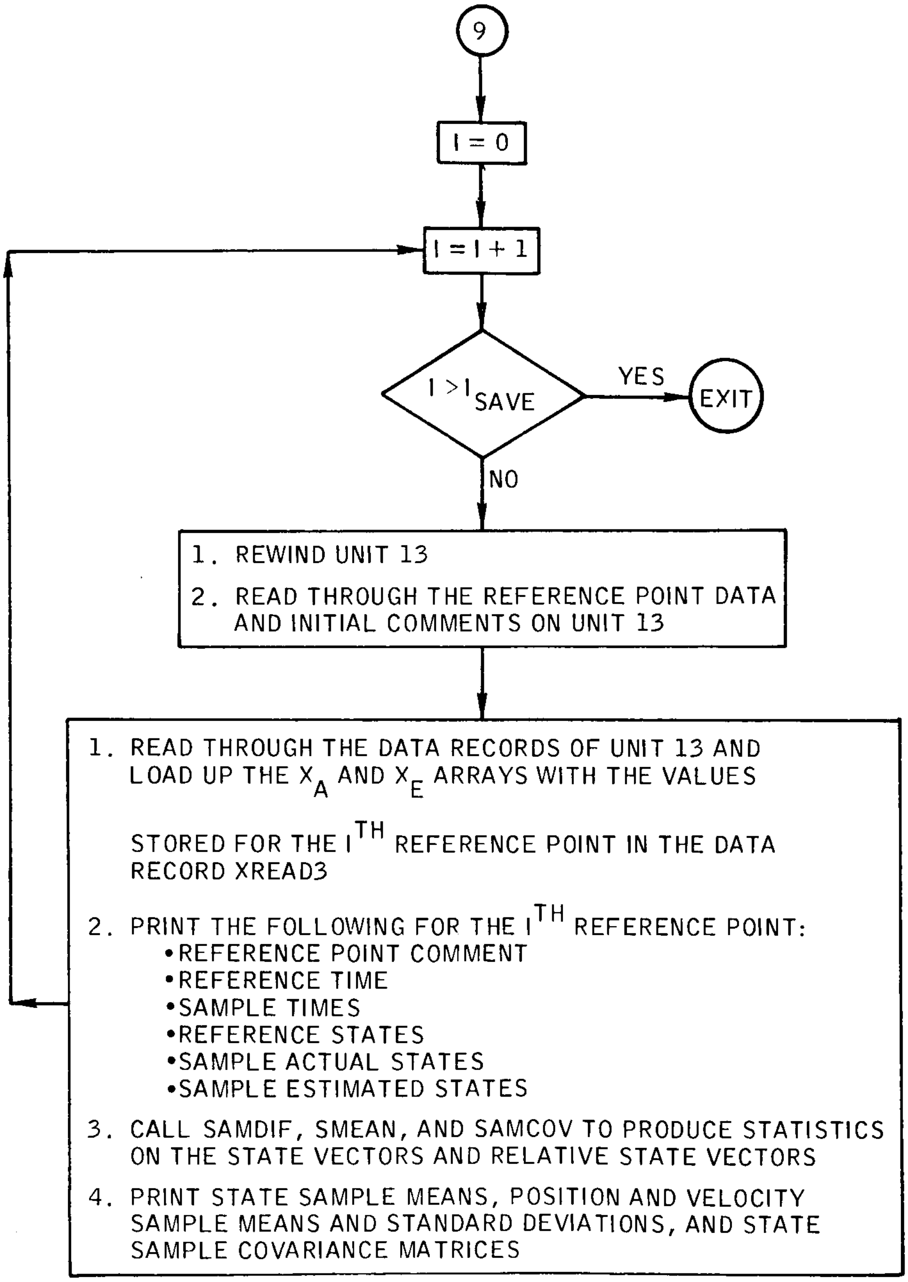
Page 5 of 7

Figure B-69.- Continued.


8
$K_{XTR} > 0$
NO
YES
CALL SAMSIG TO PRODUCE STATISTICS FOR THE XTRA DATA ARRAY; I.E., FOR THE XREAD4 DATA RECORD
PRINT THE XTRA ARRAY AND THE ASSOCIATED STATISTICS
$I_{SAM6} > 0$
NO
YES
I = 0
I = I + 1
$I > I_{SAVE}$
YES
9/7
NO
$I_{SAM}(I) \geq 6$
NO
YES
1. REWIND UNIT 13
2. READ THROUGH THE DATA RECORDS ON UNIT 13 AND LOAD UP A TABLE WITH THE RANGE AND RANGE RATE DATA STORED FOR THE ITH REFERENCE POINT
3. CALL SAMSIG TO PRODUCE RANGE AND RANGE RATE STATISTICS
4. PRINT RANGE AND RANGE RATE DATA TABLE AND RANGE AND RANGE RATE STATISTICS

Page 6 of 7

Figure B-69.- Continued.



1. REWIND UNIT 13
2. READ THROUGH THE REFERENCE POINT DATA AND INITIAL COMMENTS ON UNIT 13

1. READ THROUGH THE DATA RECORDS OF UNIT 13 AND LOAD UP THE X_A AND X_E ARRAYS WITH THE VALUES STORED FOR THE I^{TH} REFERENCE POINT IN THE DATA RECORD XREAD3
2. PRINT THE FOLLOWING FOR THE I^{TH} REFERENCE POINT:
   - REFERENCE POINT COMMENT
   - REFERENCE TIME
   - SAMPLE TIMES
   - REFERENCE STATES
   - SAMPLE ACTUAL STATES
   - SAMPLE ESTIMATED STATES
3. CALL SAMDIF, SMEAN, AND SAMCOV TO PRODUCE STATISTICS ON THE STATE VECTORS AND RELATIVE STATE VECTORS
4. PRINT STATE SAMPLE MEANS, POSITION AND VELOCITY SAMPLE MEANS AND STANDARD DEVIATIONS, AND STATE SAMPLE COVARIANCE MATRICES

Figure B-69.- Concluded.

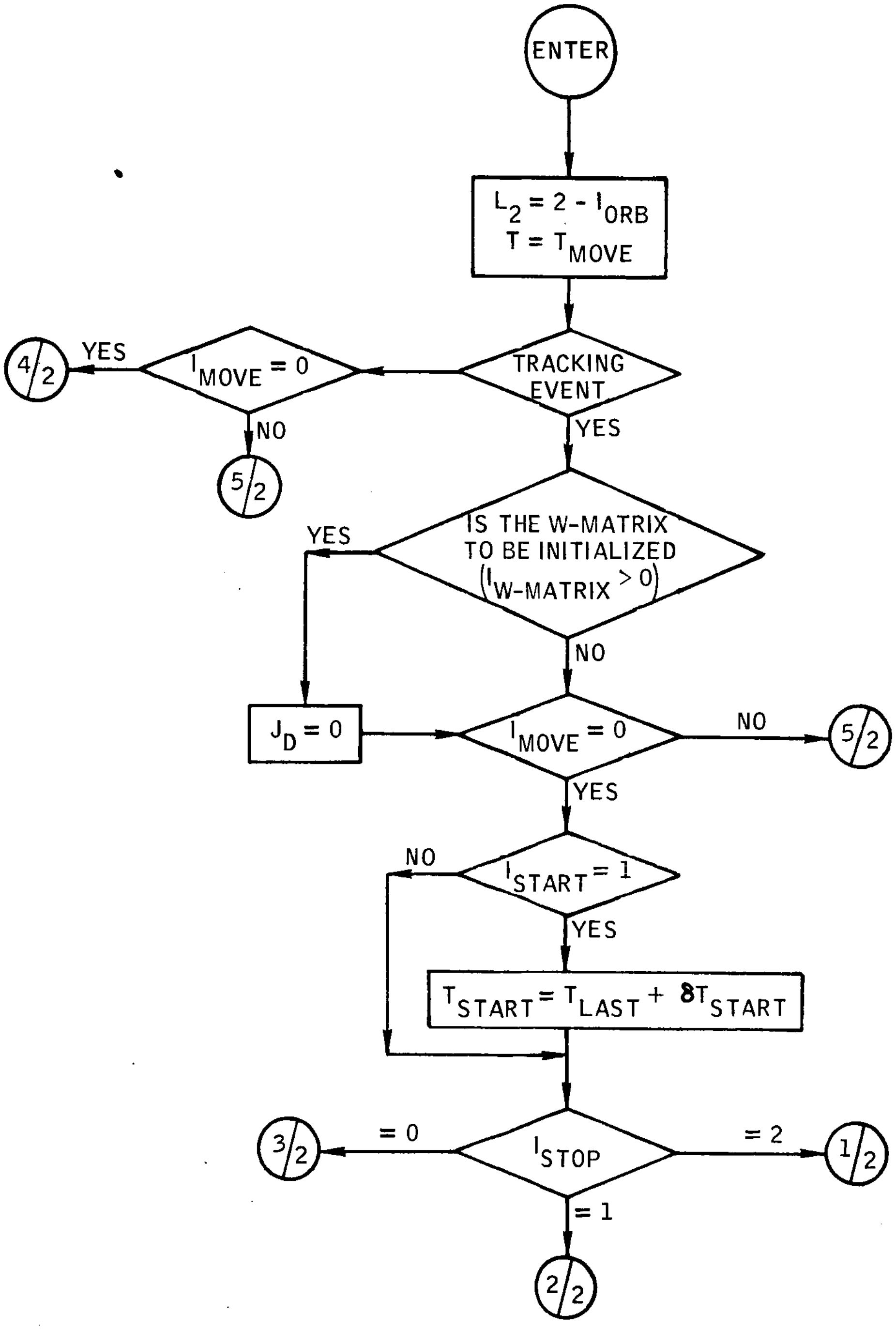
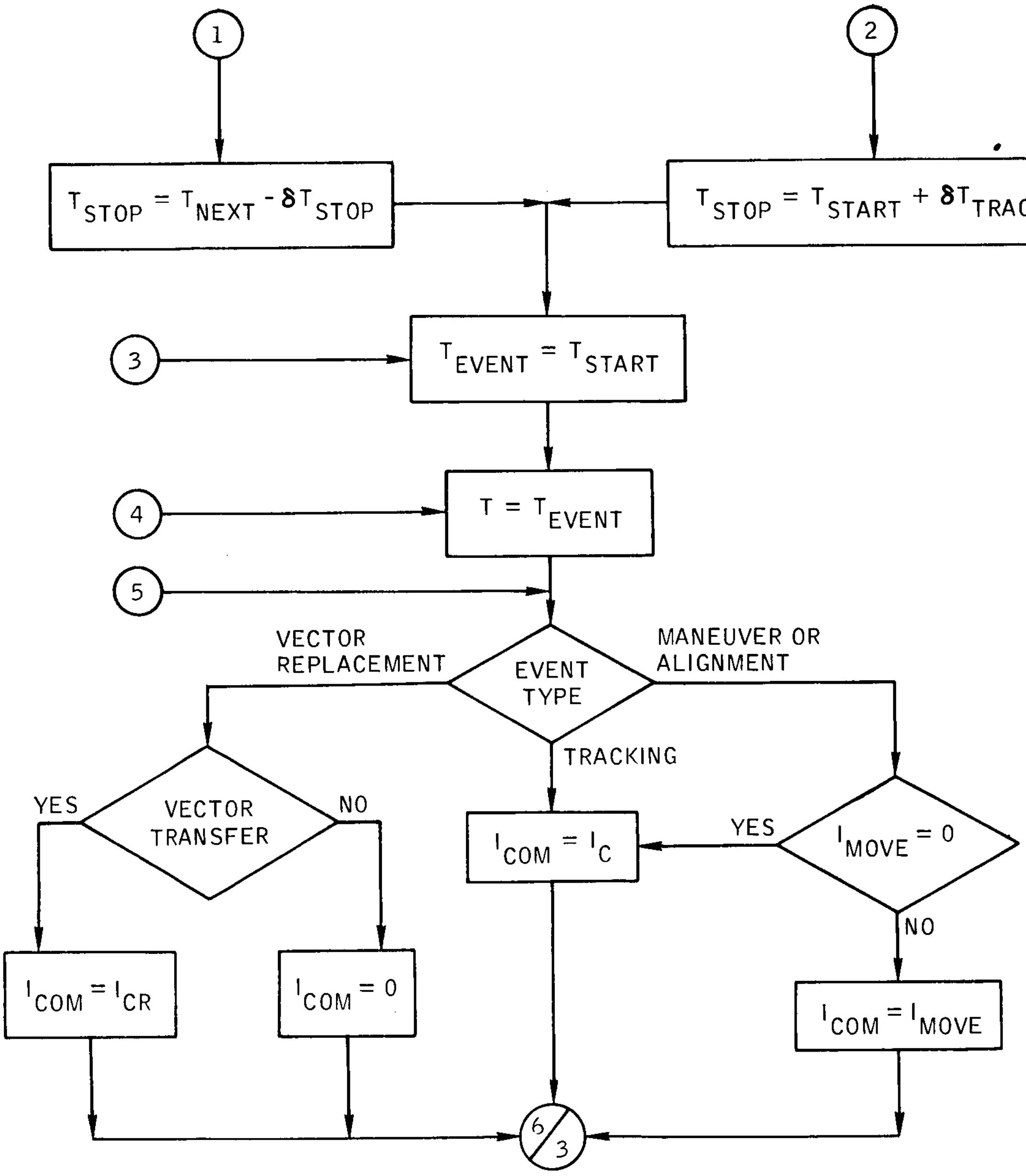
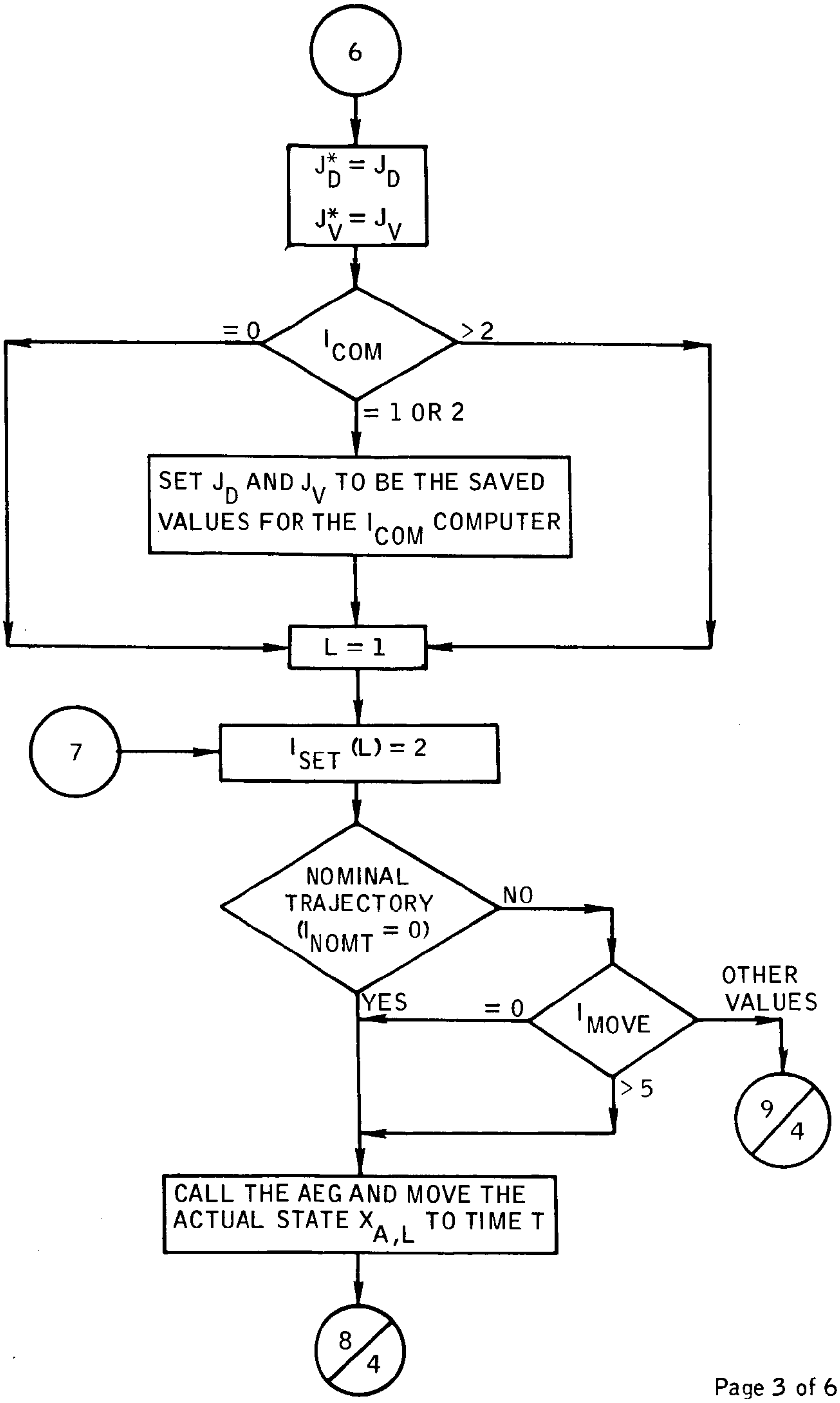


Figure B-70.- Flow chart of subroutine PROPAG. Page 1 of 6



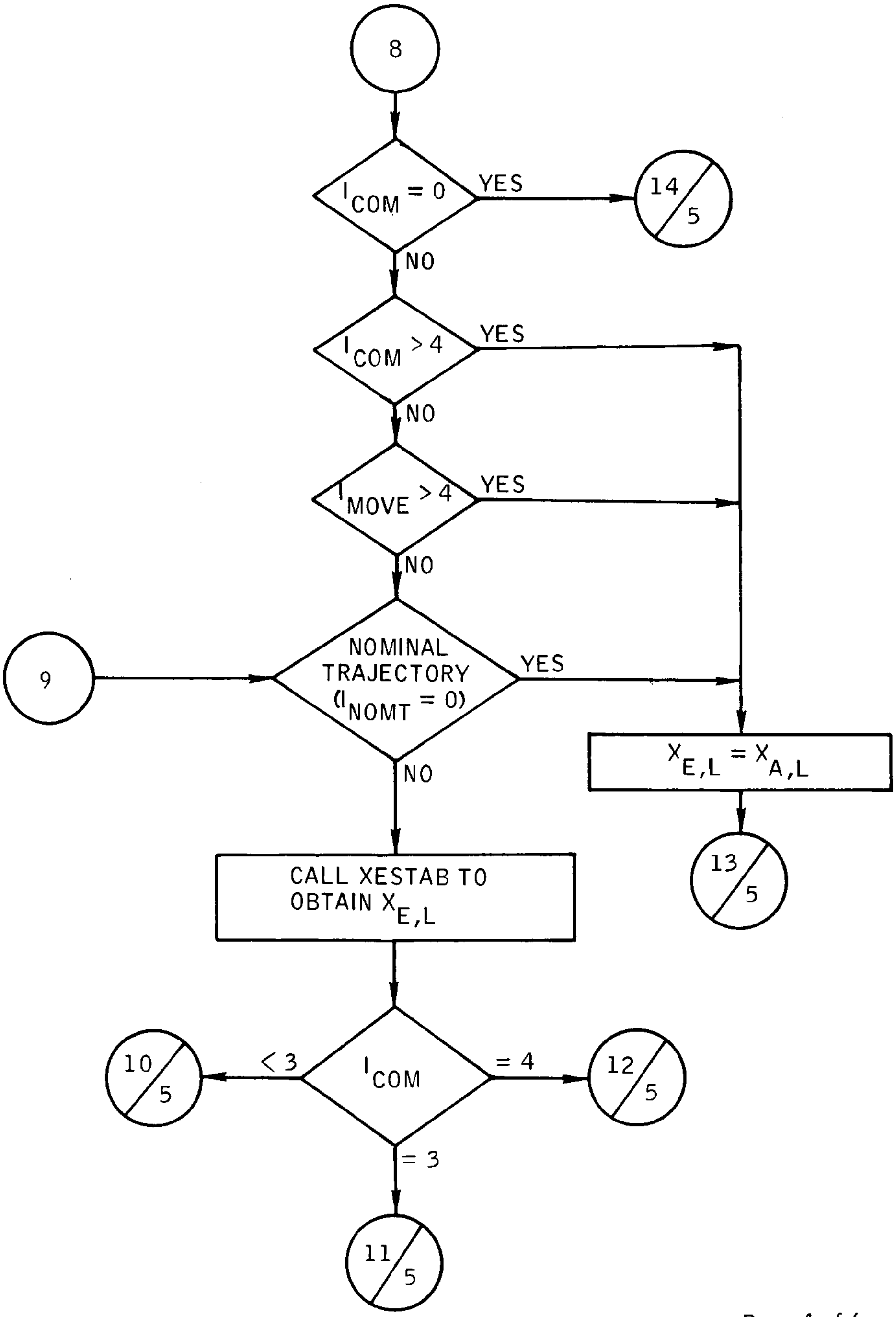
Page 2 of 6

Figure B-70.- Continued.


6
$J_D^* = J_D$
$J_V^* = J_V$
I_{COM}
= 0
> 2
= 1 OR 2
SET J_D AND J_V TO BE THE SAVED VALUES FOR THE I_{COM} COMPUTER
L = 1
7
I_{SET} (L) = 2
NOMINAL TRAJECTORY (I_{NOMT} = 0)
NO
YES
I_{MOVE}
= 0
OTHER VALUES
> 5
9/4
CALL THE AEG AND MOVE THE ACTUAL STATE $X_{A,L}$ TO TIME T
8/4

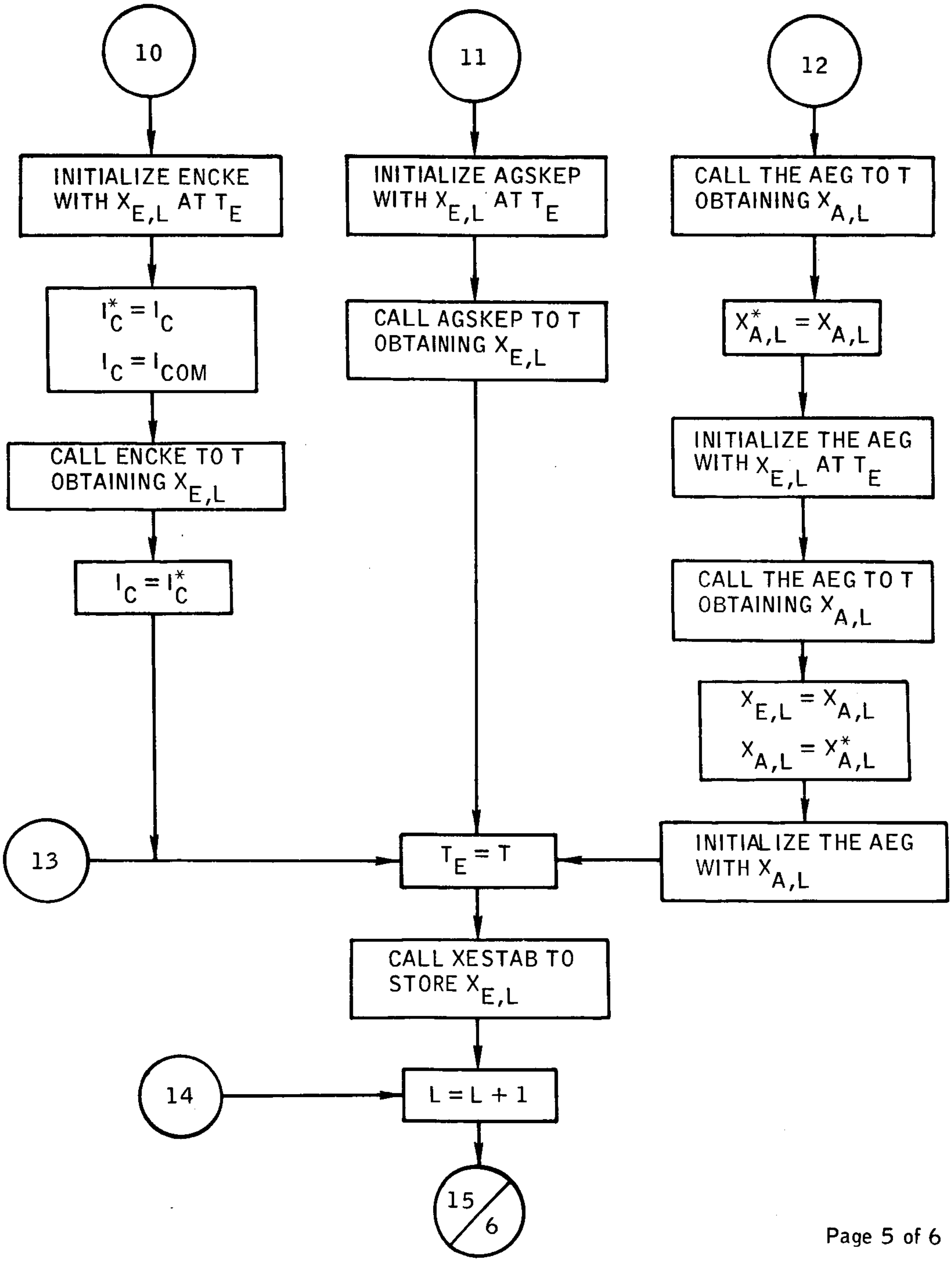
Page 3 of 6

Figure B-70.- Continued.



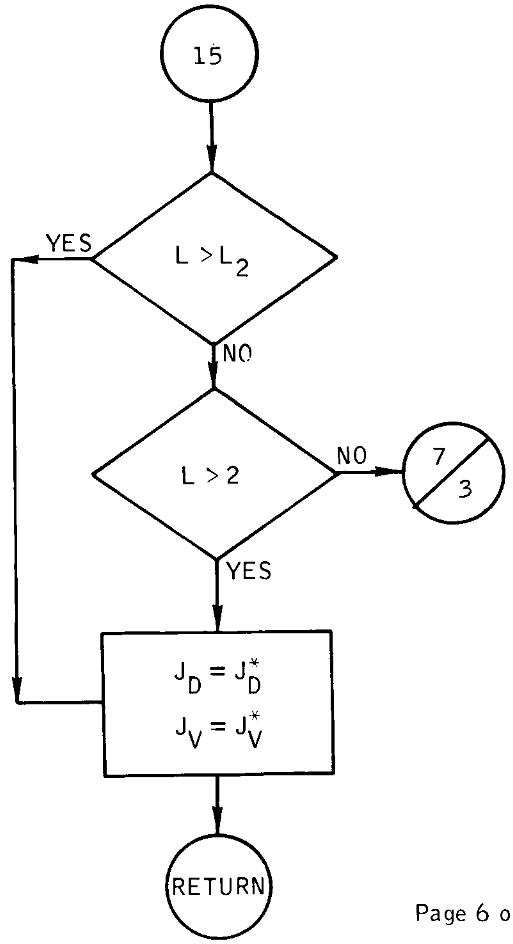
Page 4 of 6

Figure B-70. - Continued.


10
11
12
INITIALIZE ENCKE WITH $X_{E,L}$ AT T_E
INITIALIZE AGSKEP WITH $X_{E,L}$ AT T_E
CALL THE AEG TO T OBTAINING $X_{A,L}$
$I_C^* = I_C$
$I_C = I_{COM}$
CALL AGSKEP TO T OBTAINING $X_{E,L}$
$X_{A,L}^* = X_{A,L}$
CALL ENCKE TO T OBTAINING $X_{E,L}$
INITIALIZE THE AEG WITH $X_{E,L}$ AT T_E
$I_C = I_C^*$
CALL THE AEG TO T OBTAINING $X_{A,L}$
$X_{E,L} = X_{A,L}$
$X_{A,L} = X_{A,L}^*$
13
$T_E = T$
INITIALIZE THE AEG WITH $X_{A,L}$
CALL XESTAB TO STORE $X_{E,L}$
14
$L = L + 1$
15/6

Page 5 of 6

Figure B-70.- Continued.



Page 6 of 6

Figure B-70.- Concluded.

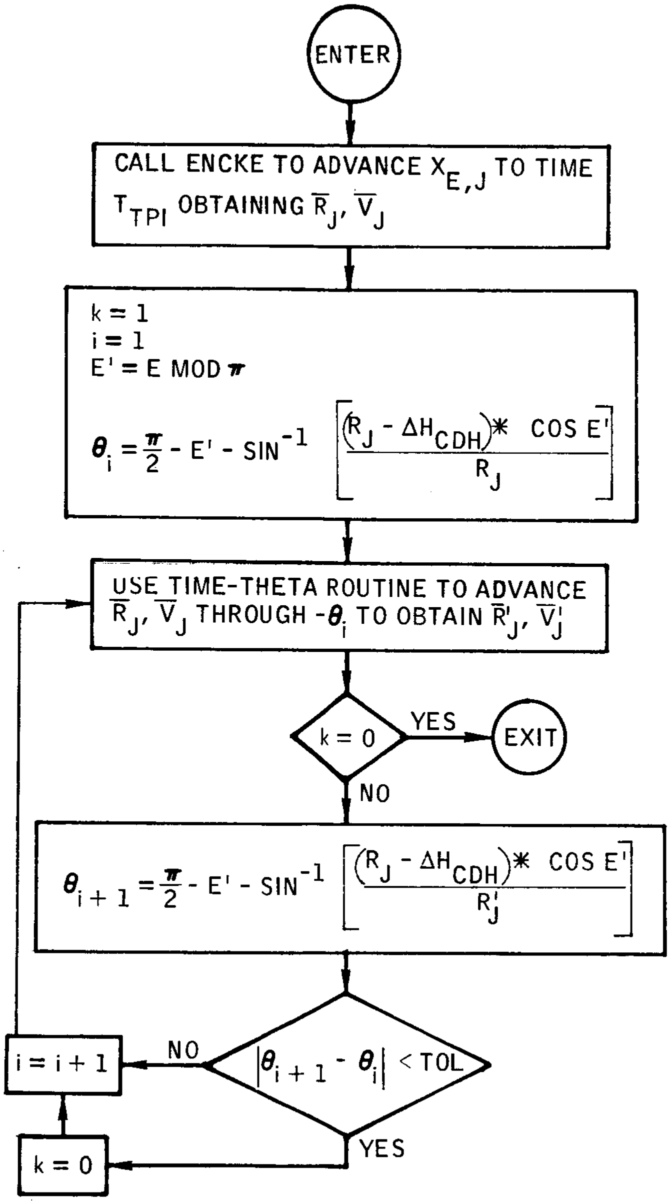
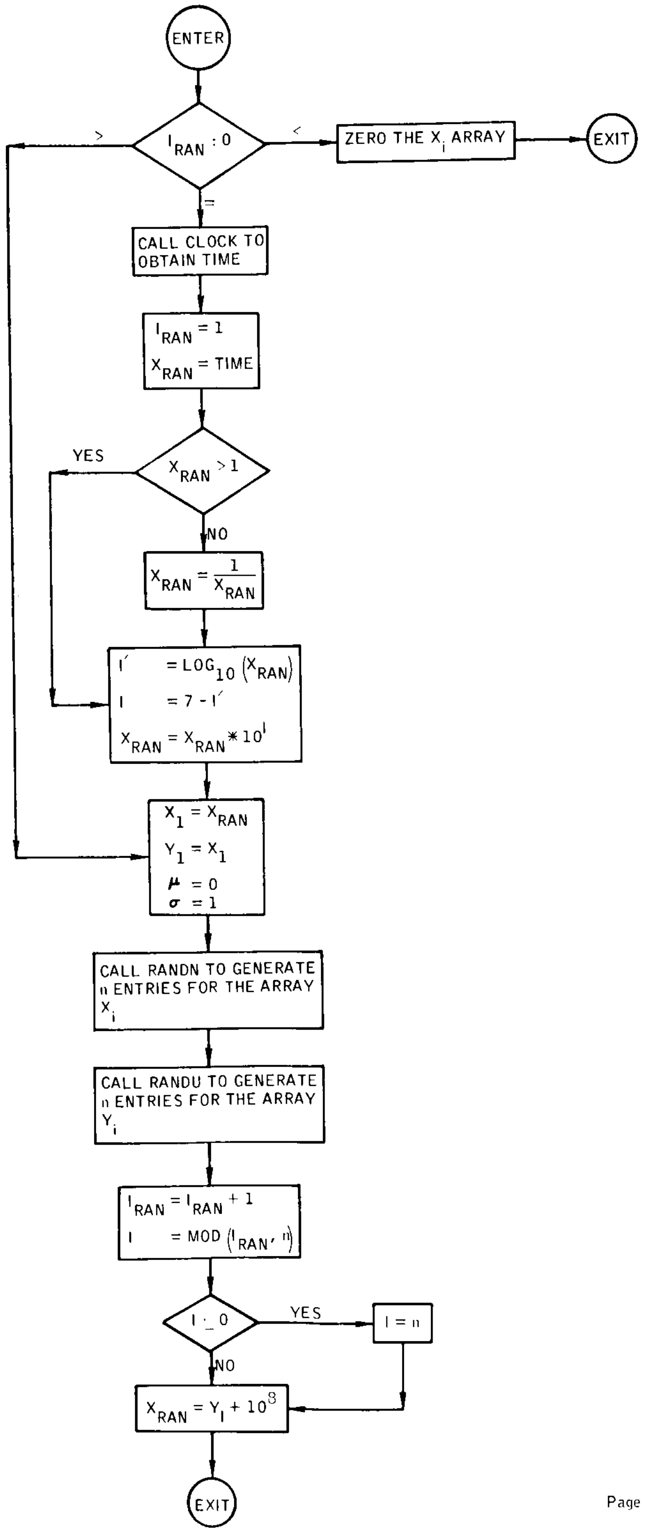

ENTER
CALL ENCKE TO ADVANCE $X_{E,J}$ TO TIME T_{TPI} OBTAINING $\overline{R}_J$, $\overline{V}_J$
$k = 1$
$i = 1$
$E' = E \text{ MOD } \pi$
$\theta_i = \frac{\pi}{2} - E' - \text{SIN}^{-1} \left[\frac{(R_J - \Delta H_{CDH})* \text{ COS } E'}{R_J} \right]$
USE TIME-THETA ROUTINE TO ADVANCE $\overline{R}_J$, $\overline{V}_J$ THROUGH $-\theta_i$ TO OBTAIN $\overline{R}'_J$, $\overline{V}'_J$
$k = 0$
YES
EXIT
NO
$\theta_{i+1} = \frac{\pi}{2} - E' - \text{SIN}^{-1} \left[\frac{(R_J - \Delta H_{CDH})* \text{ COS } E'}{R'_J} \right]$
$|\theta_{i+1} - \theta_i| < \text{TOL}$
NO
$i = i + 1$
YES
$k = 0$

Figure B-71.- Flow chart of subroutine QRDTPI.



Page 1 of 1

Figure B-72.- Flow chart of subroutine RANNO.

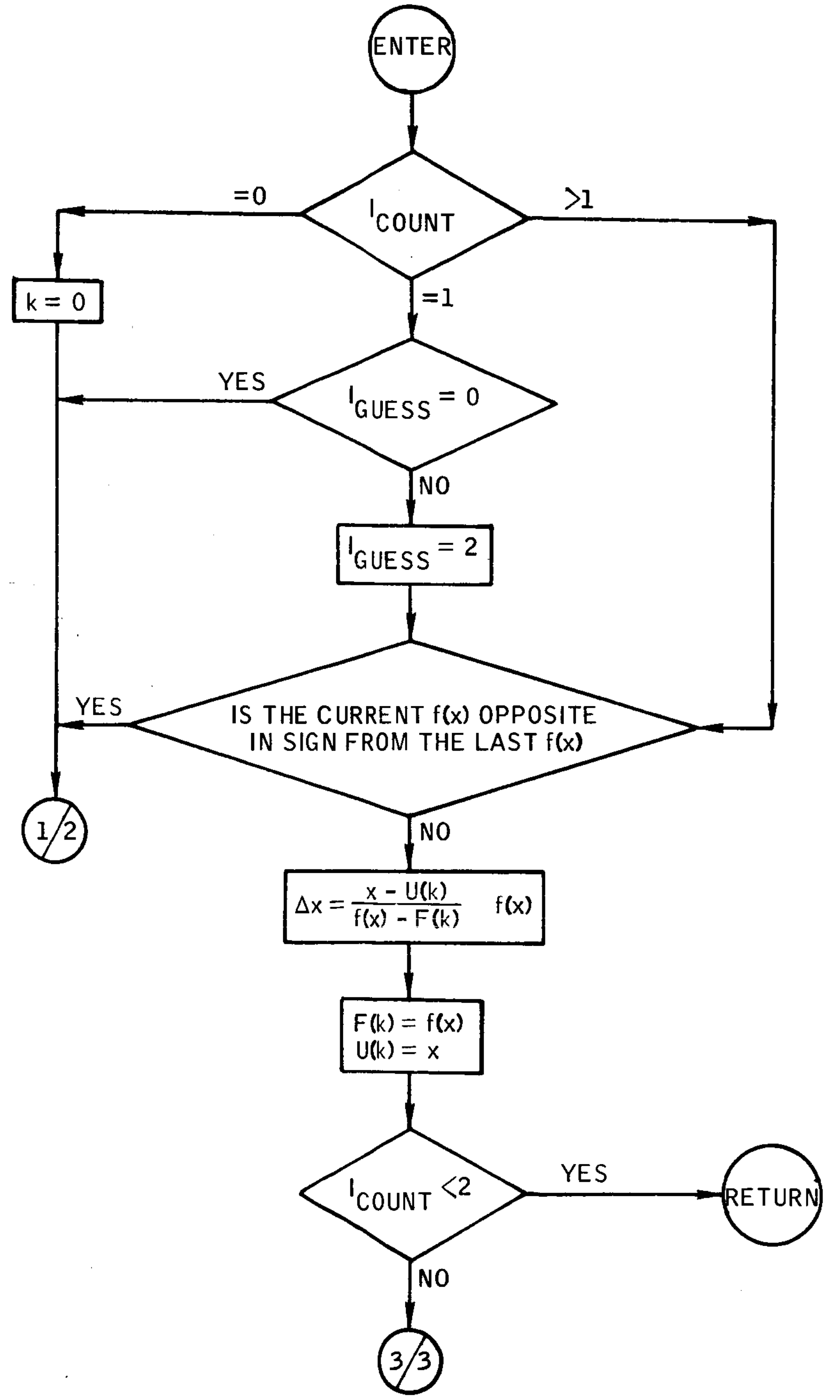


Figure B-73.- Flow chart for subroutine REGFAL.

Page 1 of 3

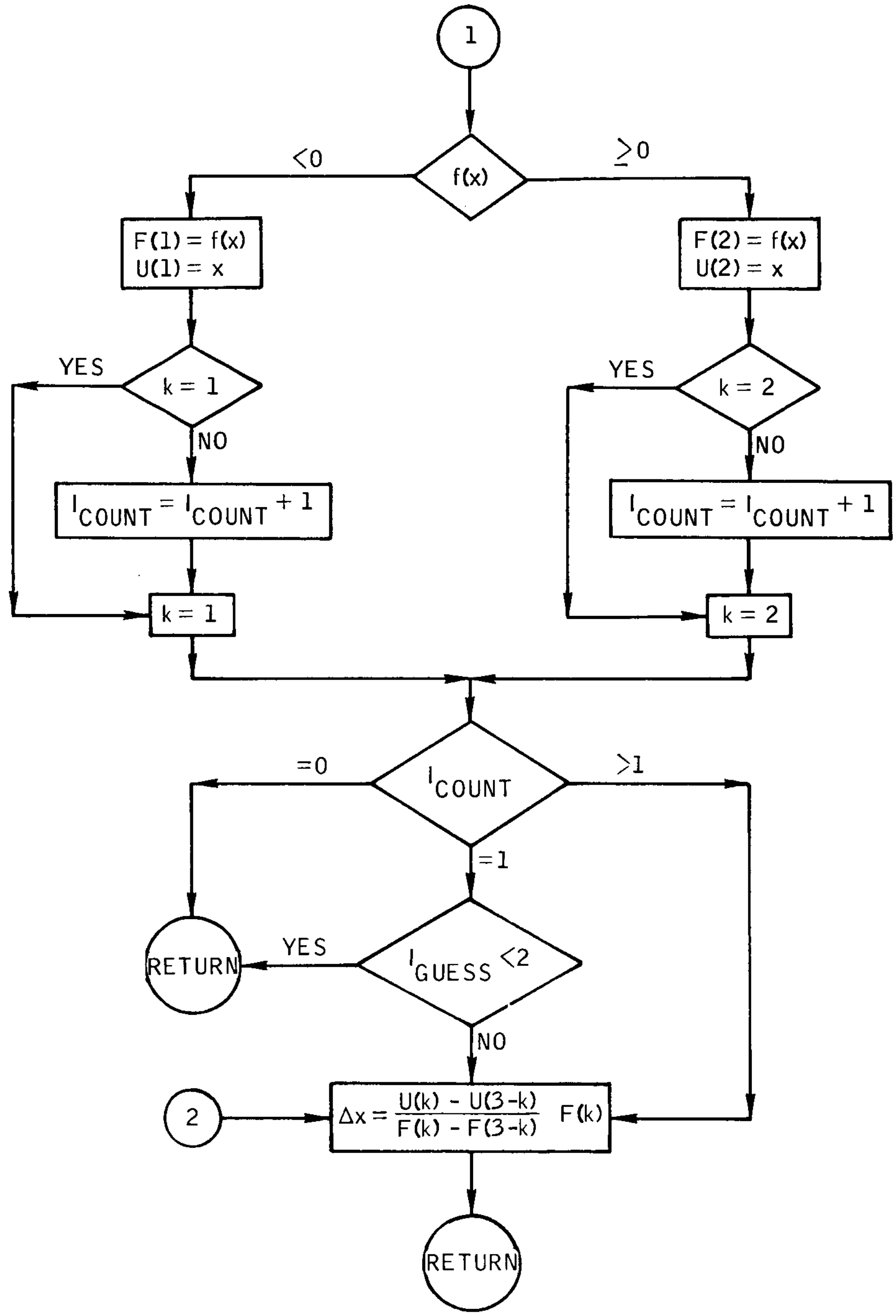


Figure B-73.- Continued.

Page 2 of 3

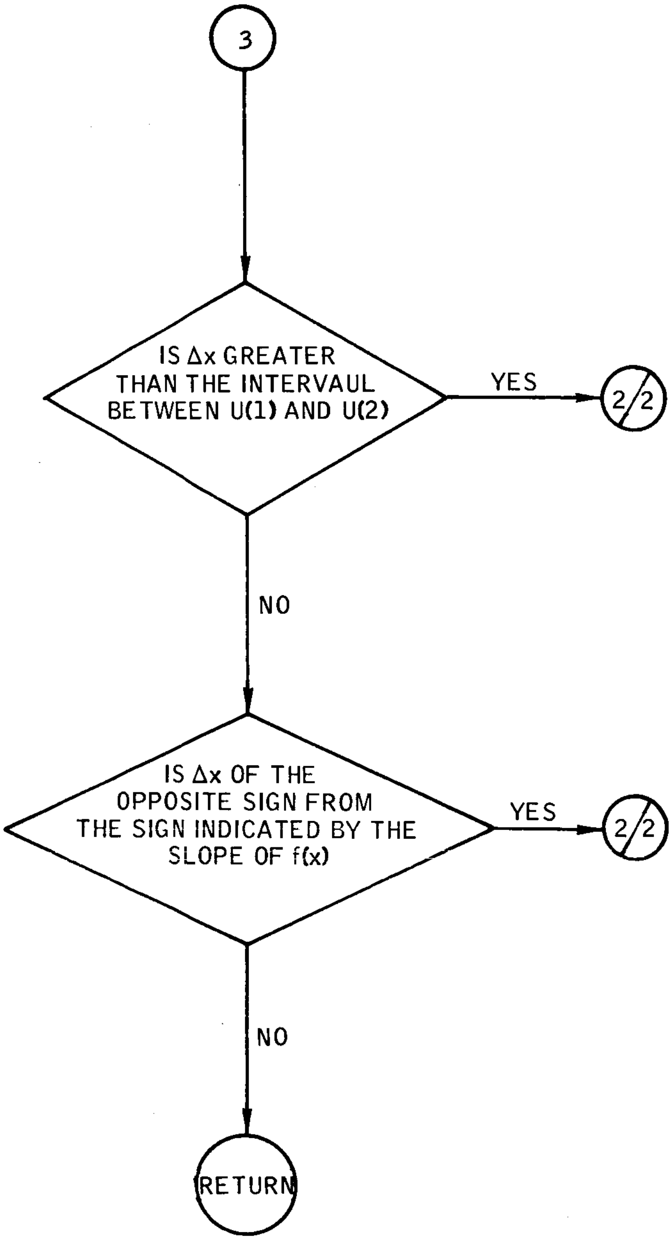
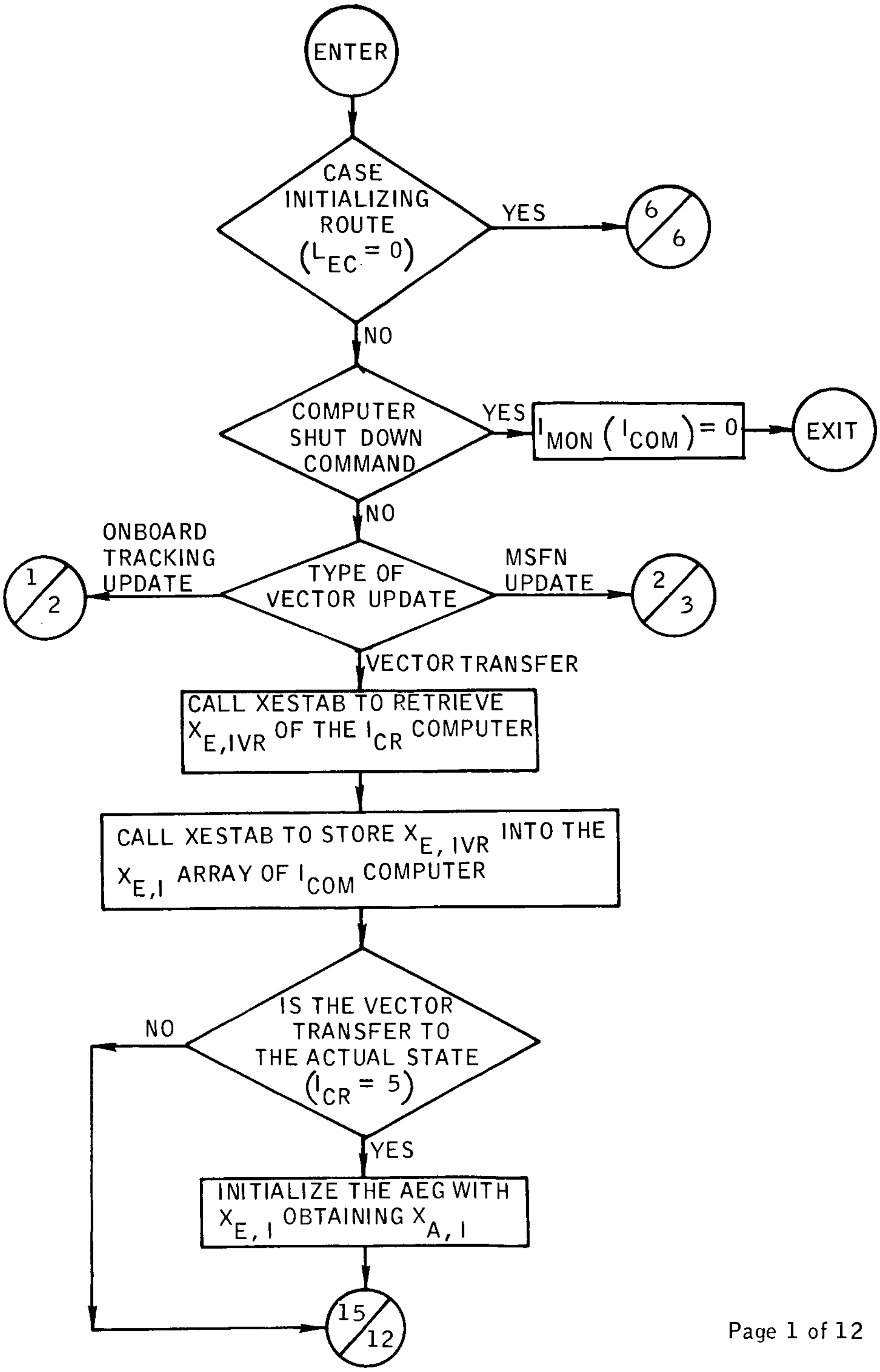

3
IS Δx GREATER THAN THE INTERVAUL BETWEEN U(1) AND U(2)
YES
2/2
NO
IS Δx OF THE OPPOSITE SIGN FROM THE SIGN INDICATED BY THE SLOPE OF f(x)
YES
2/2
NO
RETURN

Figure B-73.- Concluded.

Page 3 of 3


ENTER
CASE INITIALIZING ROUTE (L_{EC} = 0)
YES
6/6
NO
COMPUTER SHUT DOWN COMMAND
YES
$I_{MON}(I_{COM})$ = 0
EXIT
NO
ONBOARD TRACKING UPDATE
1/2
TYPE OF VECTOR UPDATE
MSFN UPDATE
2/3
VECTOR TRANSFER
CALL XESTAB TO RETRIEVE $X_{E,IVR}$ OF THE I_{CR} COMPUTER
CALL XESTAB TO STORE $X_{E,IVR}$ INTO THE $X_{E,I}$ ARRAY OF I_{COM} COMPUTER
IS THE VECTOR TRANSFER TO THE ACTUAL STATE (I_{CR} = 5)
NO
YES
INITIALIZE THE AEG WITH $X_{E,I}$ OBTAINING $X_{A,I}$
15/12

Page 1 of 12

Figure B-74.- Flow chart of subroutine REPLAC.

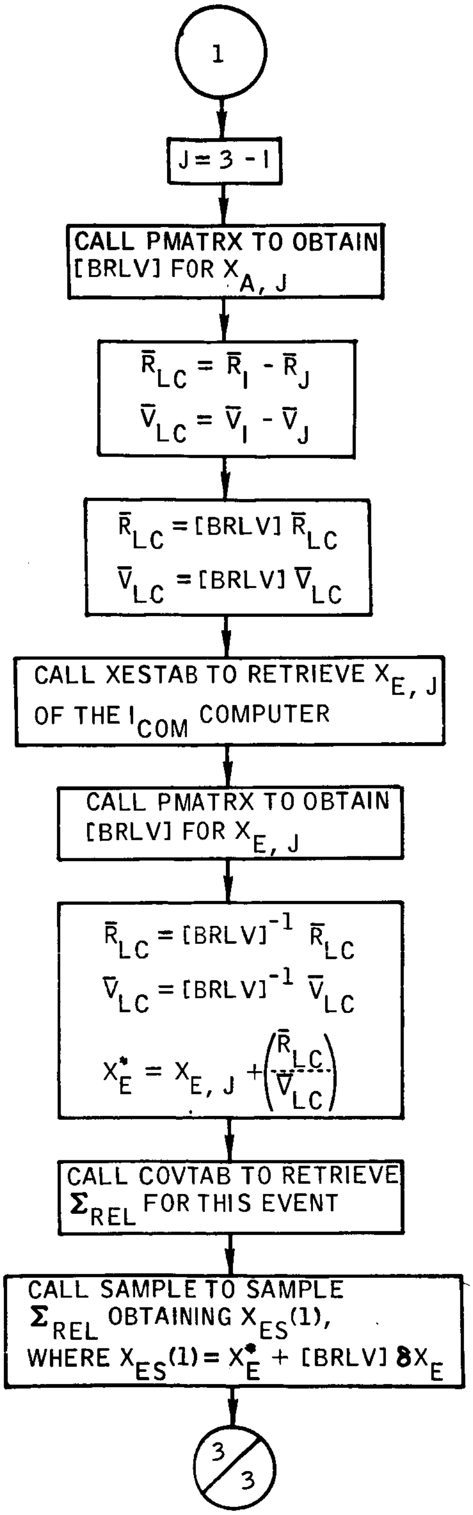
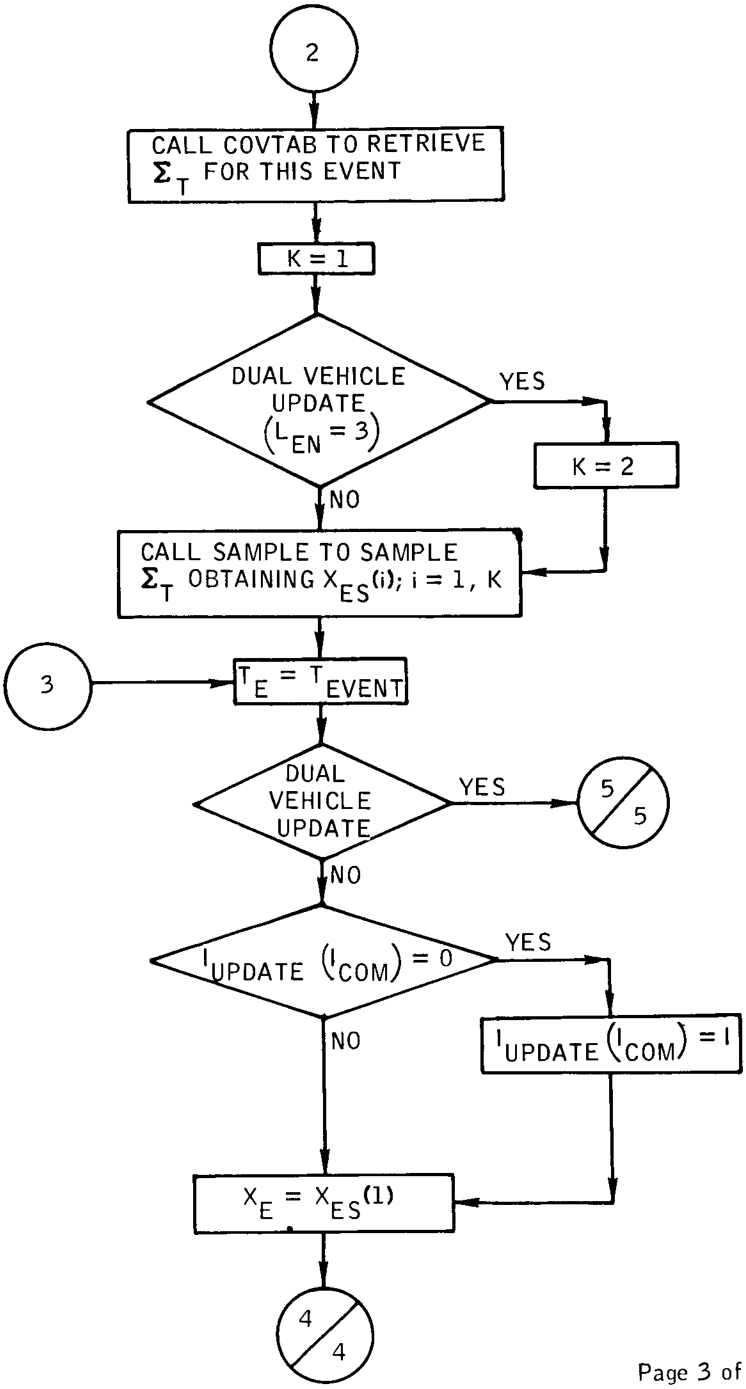
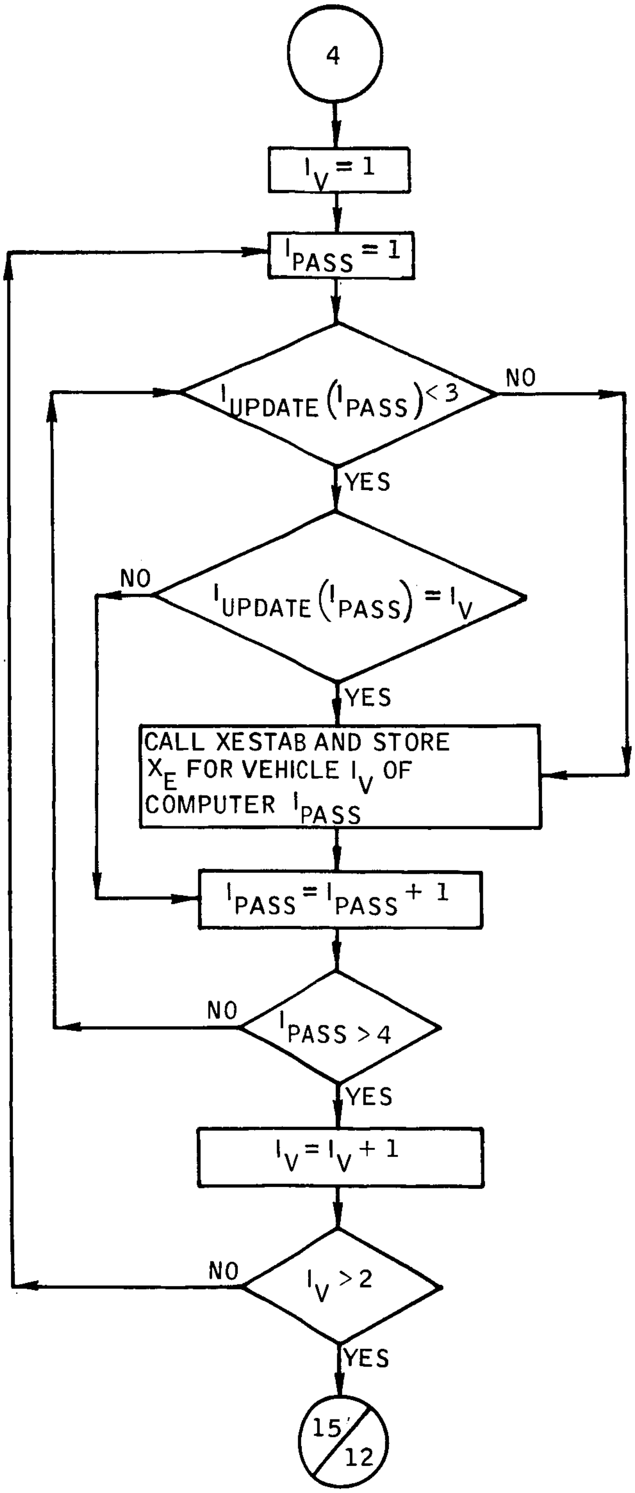

1
J = 3 - I
CALL PMATRX TO OBTAIN [BRLV] FOR $X_{A, J}$
$\bar{R}_{LC} = \bar{R}_I - \bar{R}_J$
$\bar{V}_{LC} = \bar{V}_I - \bar{V}_J$
$\bar{R}_{LC} = [BRLV]\,\bar{R}_{LC}$
$\bar{V}_{LC} = [BRLV]\,\bar{V}_{LC}$
CALL XESTAB TO RETRIEVE $X_{E, J}$ OF THE I_{COM} COMPUTER
CALL PMATRX TO OBTAIN [BRLV] FOR $X_{E, J}$
$\bar{R}_{LC} = [BRLV]^{-1}\,\bar{R}_{LC}$
$\bar{V}_{LC} = [BRLV]^{-1}\,\bar{V}_{LC}$
$X_E^* = X_{E, J} + \left(\frac{\bar{R}_{LC}}{\bar{V}_{LC}}\right)$
CALL COVTAB TO RETRIEVE Σ_{REL} FOR THIS EVENT
CALL SAMPLE TO SAMPLE Σ_{REL} OBTAINING $X_{ES}(1)$, WHERE $X_{ES}(1) = X_E^* + [BRLV]\,\delta X_E$
3
3

Figure B-74. - Continued.


2
CALL COVTAB TO RETRIEVE Σ_T FOR THIS EVENT
K = 1
DUAL VEHICLE UPDATE $(L_{EN} = 3)$
YES
K = 2
NO
CALL SAMPLE TO SAMPLE Σ_T OBTAINING $X_{ES}(i)$; i = 1, K
3
$T_E = T_{EVENT}$
DUAL VEHICLE UPDATE
YES
5/5
NO
$I_{UPDATE}(I_{COM}) = 0$
YES
NO
$I_{UPDATE}(I_{COM}) = 1$
$X_E = X_{ES}(1)$
4/4

Page 3 of 12

Figure B-74.- Continued.



Page 4 of 12

Figure B-74.- Continued.

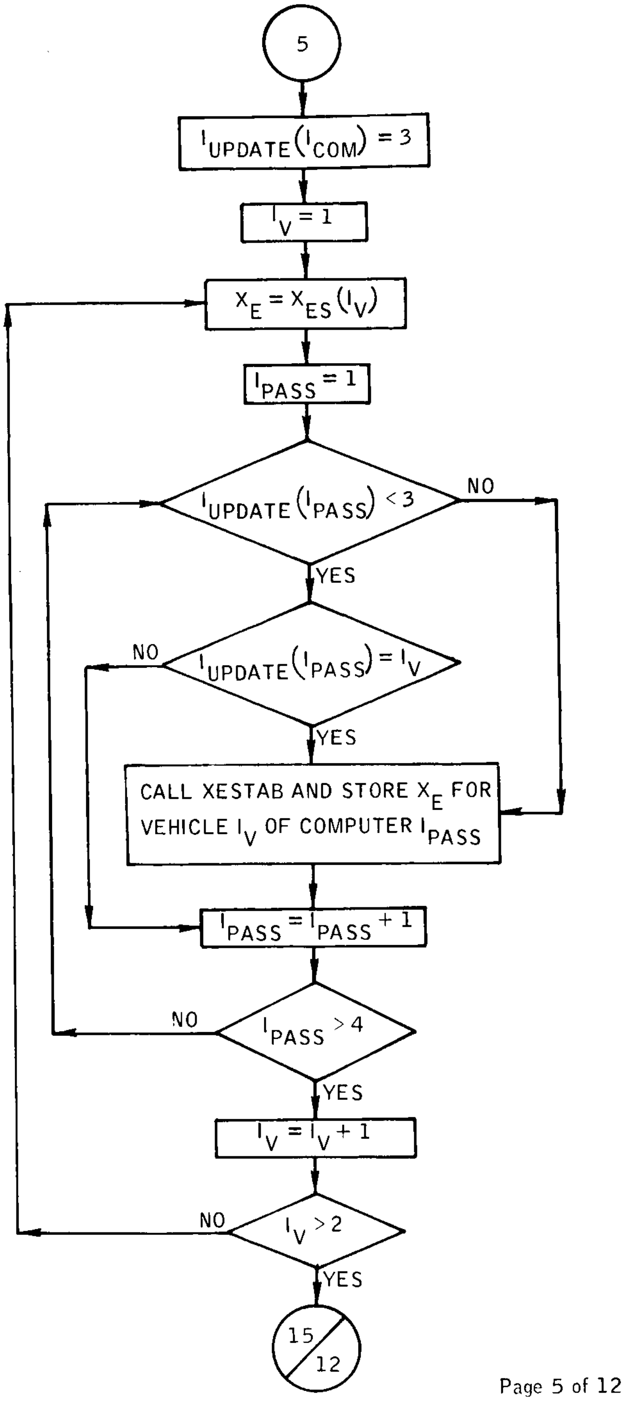


Figure B-74.- Continued.

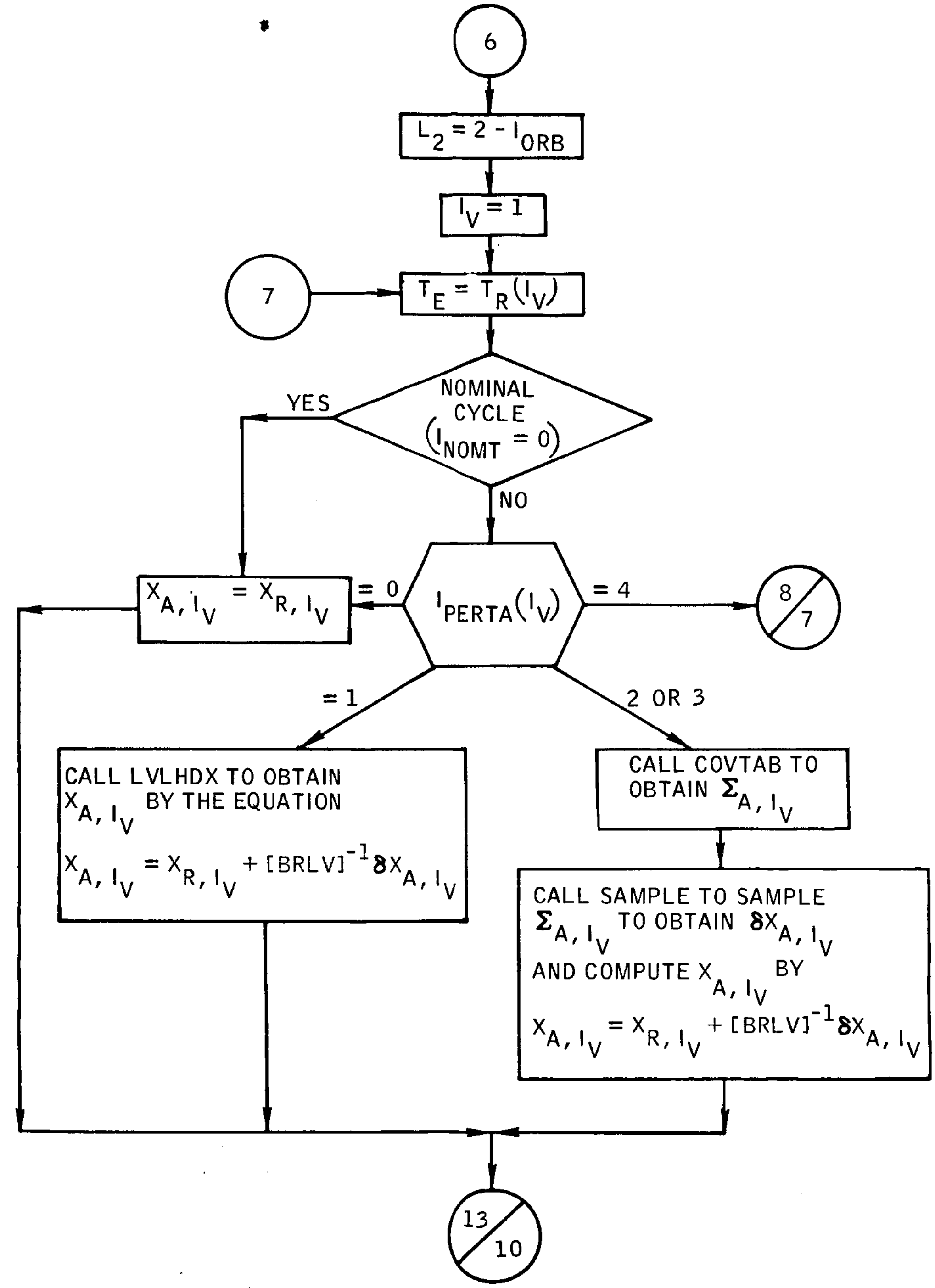

6
$L_2 = 2 - I_{ORB}$
$I_V = 1$
7
$T_E = T_R(I_V)$
NOMINAL CYCLE $(I_{NOMT} = 0)$
YES
NO
$X_{A, I_V} = X_{R, I_V}$
= 0
$I_{PERTA}(I_V)$
= 4
8/7
= 1
2 OR 3
CALL LVLHDX TO OBTAIN X_{A, I_V} BY THE EQUATION
$X_{A, I_V} = X_{R, I_V} + [BRLV]^{-1} \delta X_{A, I_V}$
CALL COVTAB TO OBTAIN Σ_{A, I_V}
CALL SAMPLE TO SAMPLE Σ_{A, I_V} TO OBTAIN $\delta X_{A, I_V}$ AND COMPUTE X_{A, I_V} BY
$X_{A, I_V} = X_{R, I_V} + [BRLV]^{-1} \delta X_{A, I_V}$
13/10

Figure B-74.- Continued.

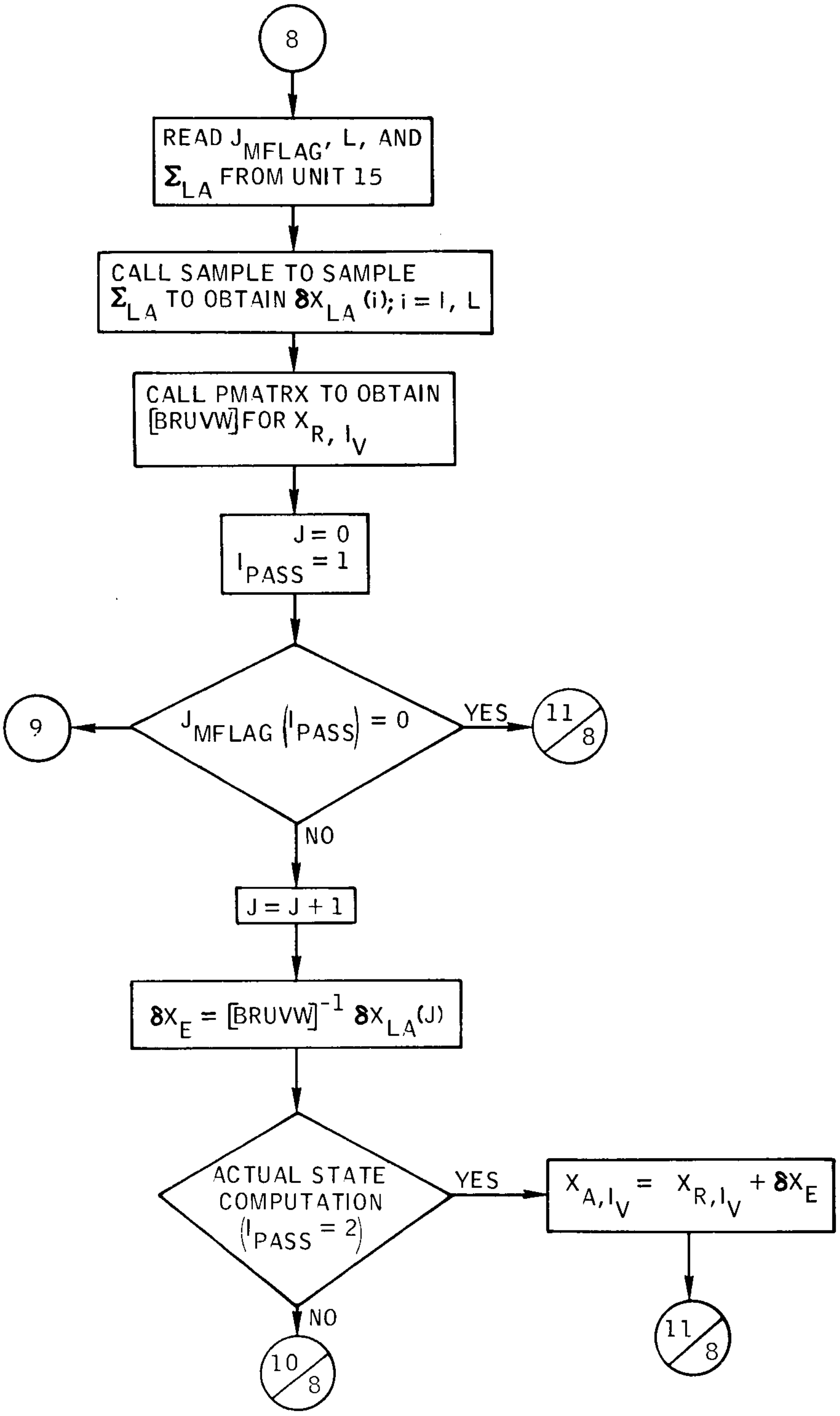

8
READ J_{MFLAG}, L, AND Σ_{LA} FROM UNIT 15
CALL SAMPLE TO SAMPLE Σ_{LA} TO OBTAIN $\delta X_{LA}(i)$; i = l, L
CALL PMATRX TO OBTAIN [BRUVW] FOR X_{R, I_V}
J = 0
$I_{PASS} = 1$
$J_{MFLAG}(I_{PASS}) = 0$
YES
11/8
9
NO
J = J + 1
$\delta X_E = [BRUVW]^{-1} \delta X_{LA}(J)$
ACTUAL STATE COMPUTATION ($I_{PASS} = 2$)
YES
$X_{A,I_V} = X_{R,I_V} + \delta X_E$
11/8
NO
10/8

Figure B-74.- Continued.

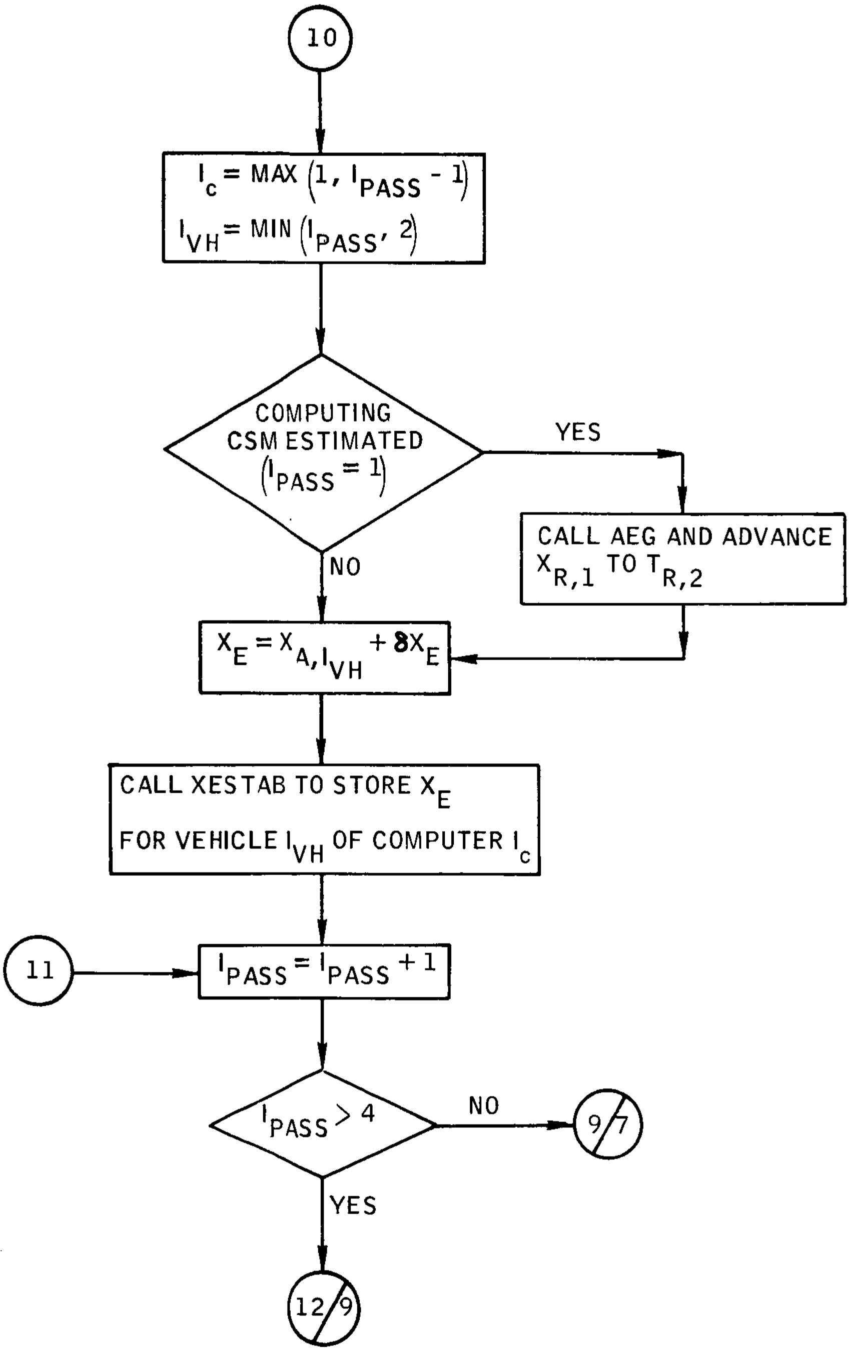
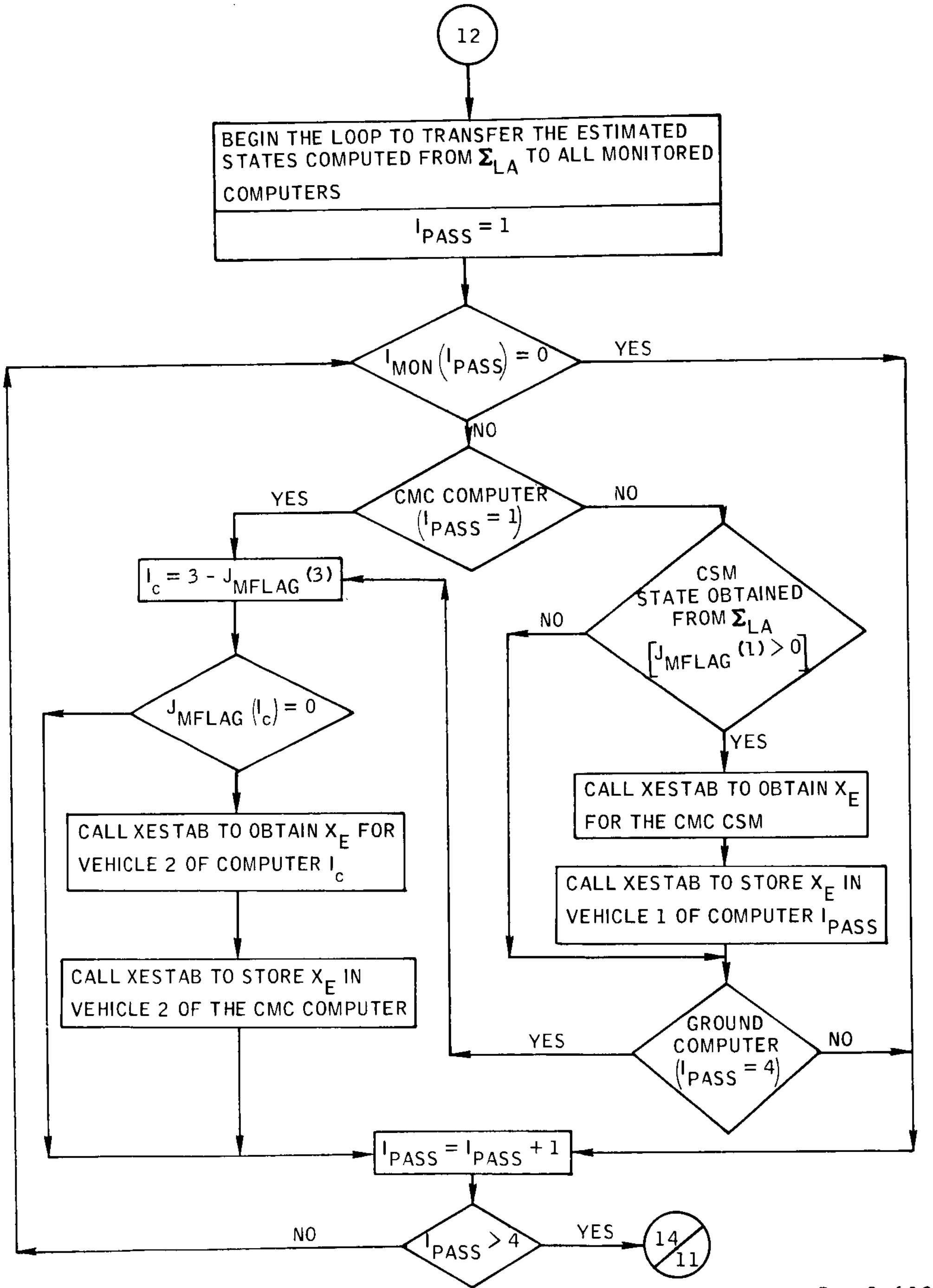

10
$I_c = MAX(1, I_{PASS} - 1)$
$I_{VH} = MIN(I_{PASS}, 2)$
COMPUTING CSM ESTIMATED $(I_{PASS} = 1)$
YES
CALL AEG AND ADVANCE $X_{R,1}$ TO $T_{R,2}$
NO
$X_E = X_{A,I_{VH}} + \delta X_E$
CALL XESTAB TO STORE X_E FOR VEHICLE I_{VH} OF COMPUTER I_c
11
$I_{PASS} = I_{PASS} + 1$
$I_{PASS} > 4$
NO
9/7
YES
12/9

Figure B-74.- Continued.


12
BEGIN THE LOOP TO TRANSFER THE ESTIMATED STATES COMPUTED FROM Σ_{LA} TO ALL MONITORED COMPUTERS
$I_{PASS} = 1$
$I_{MON}(I_{PASS}) = 0$
YES
NO
CMC COMPUTER $(I_{PASS} = 1)$
YES
NO
$I_c = 3 - J_{MFLAG}(3)$
CSM STATE OBTAINED FROM Σ_{LA} $[J_{MFLAG}(1) > 0]$
NO
$J_{MFLAG}(I_c) = 0$
YES
CALL XESTAB TO OBTAIN X_E FOR THE CMC CSM
CALL XESTAB TO OBTAIN X_E FOR VEHICLE 2 OF COMPUTER I_c
CALL XESTAB TO STORE X_E IN VEHICLE 1 OF COMPUTER I_{PASS}
CALL XESTAB TO STORE X_E IN VEHICLE 2 OF THE CMC COMPUTER
YES
GROUND COMPUTER $(I_{PASS} = 4)$
NO
$I_{PASS} = I_{PASS} + 1$
NO
$I_{PASS} > 4$
YES
14/11

Page 9 of 12

Figure B-74.- Continued.

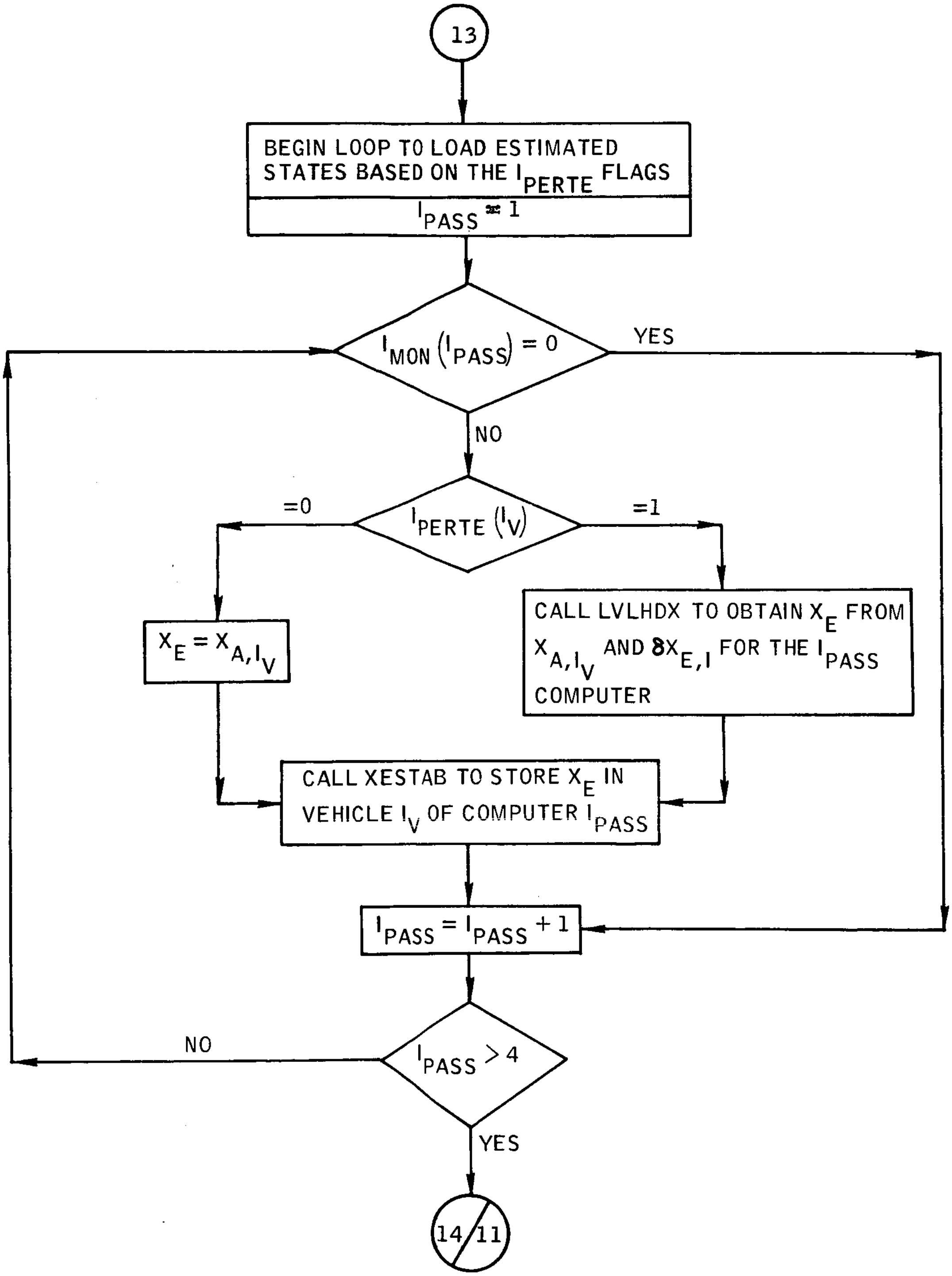

13
BEGIN LOOP TO LOAD ESTIMATED STATES BASED ON THE I_{PERTE} FLAGS
$I_{PASS} = 1$
$I_{MON}(I_{PASS}) = 0$
YES
NO
$I_{PERTE}(I_V)$
=0
=1
$X_E = X_{A,I_V}$
CALL LVLHDX TO OBTAIN X_E FROM X_{A,I_V} AND $\delta X_{E,I}$ FOR THE I_{PASS} COMPUTER
CALL XESTAB TO STORE X_E IN VEHICLE I_V OF COMPUTER I_{PASS}
$I_{PASS} = I_{PASS} + 1$
$I_{PASS} > 4$
NO
YES
14/11

Figure B-74.- Continued.

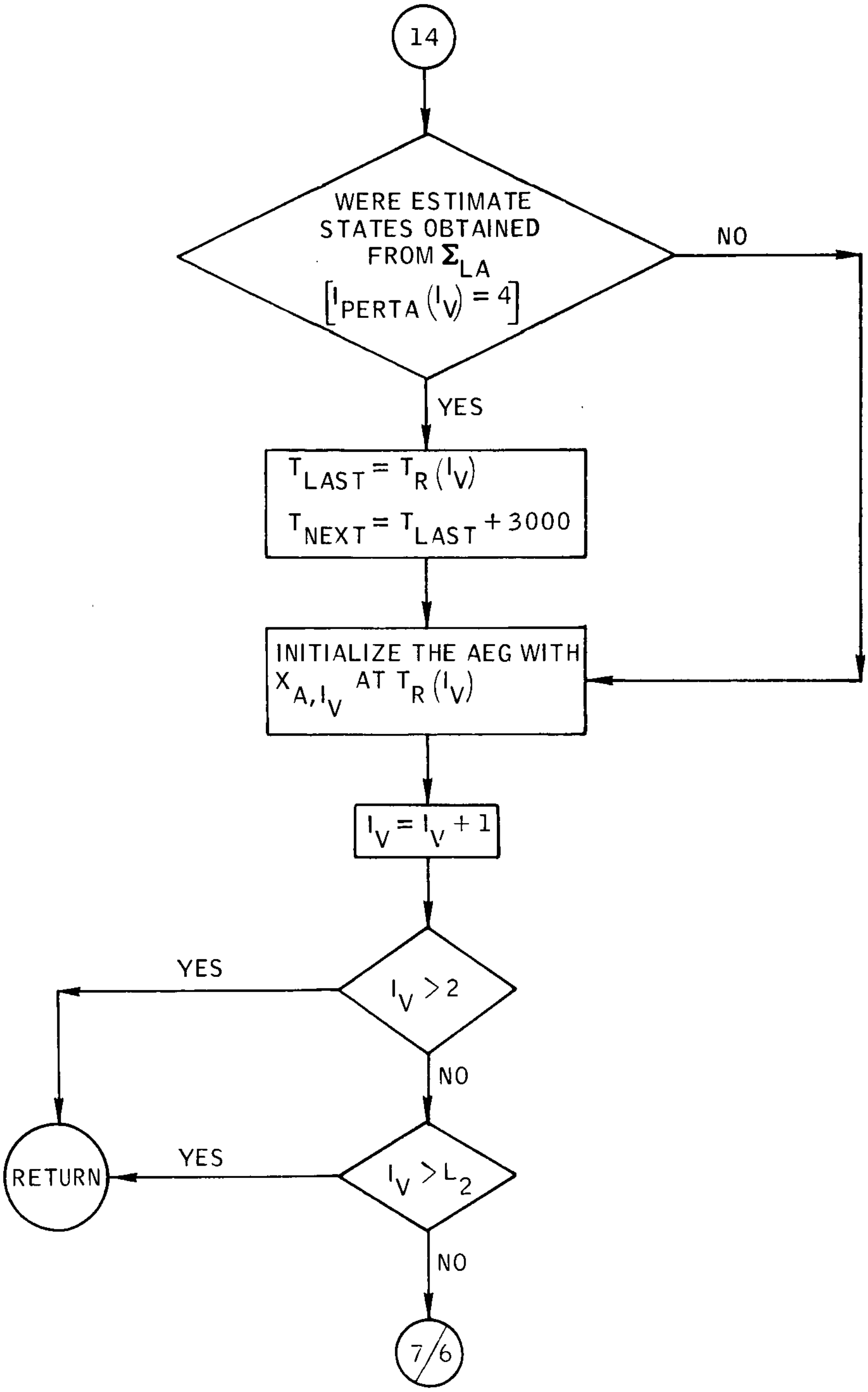

14
WERE ESTIMATE STATES OBTAINED FROM Σ_{LA}
$[I_{PERTA}(I_V) = 4]$
NO
YES
$T_{LAST} = T_R(I_V)$
$T_{NEXT} = T_{LAST} + 3000$
INITIALIZE THE AEG WITH X_{A,I_V} AT $T_R(I_V)$
$I_V = I_V + 1$
YES
$I_V > 2$
NO
YES
RETURN
$I_V > L_2$
NO
7/6

Figure 74.- Continued.

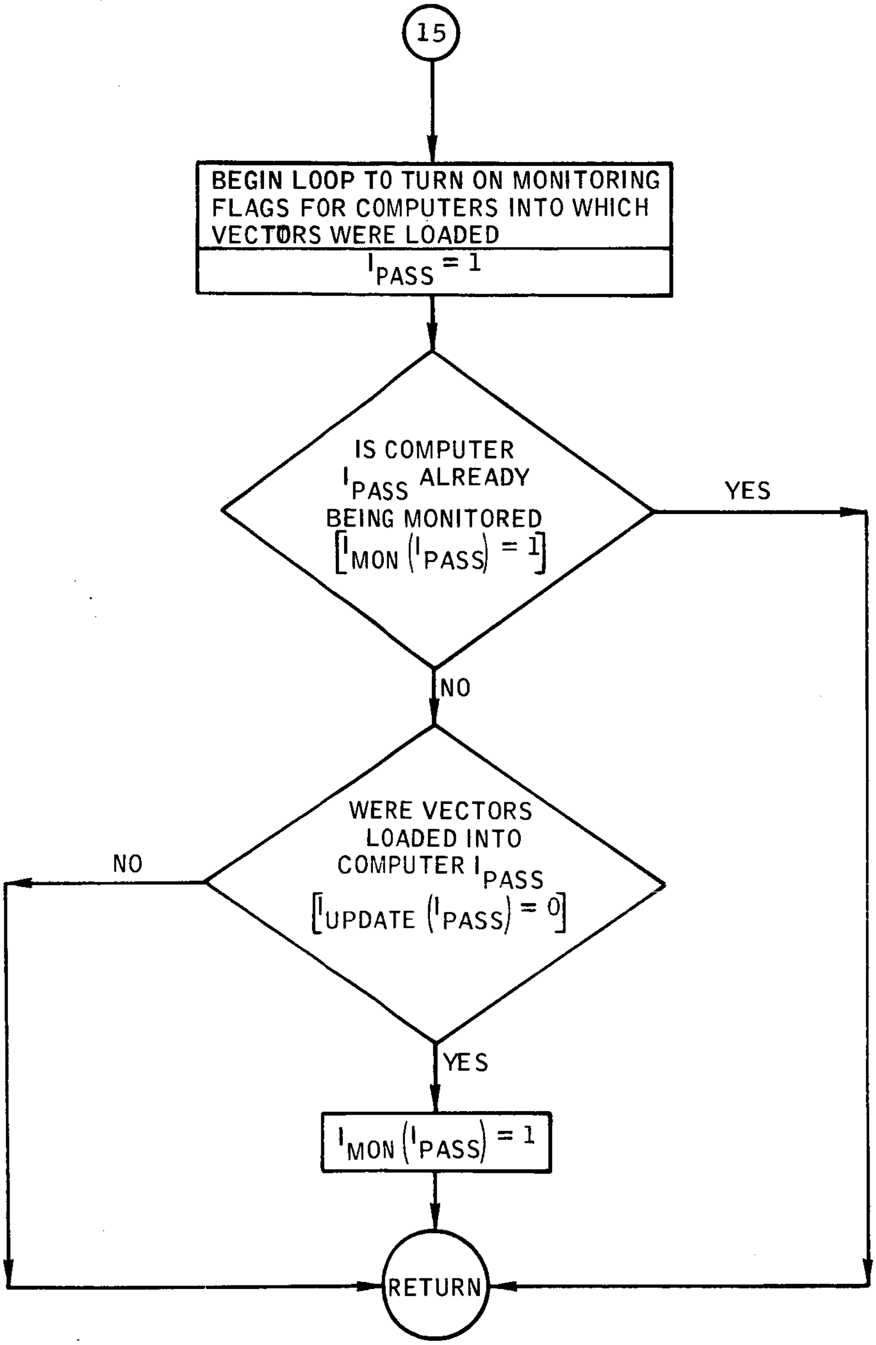
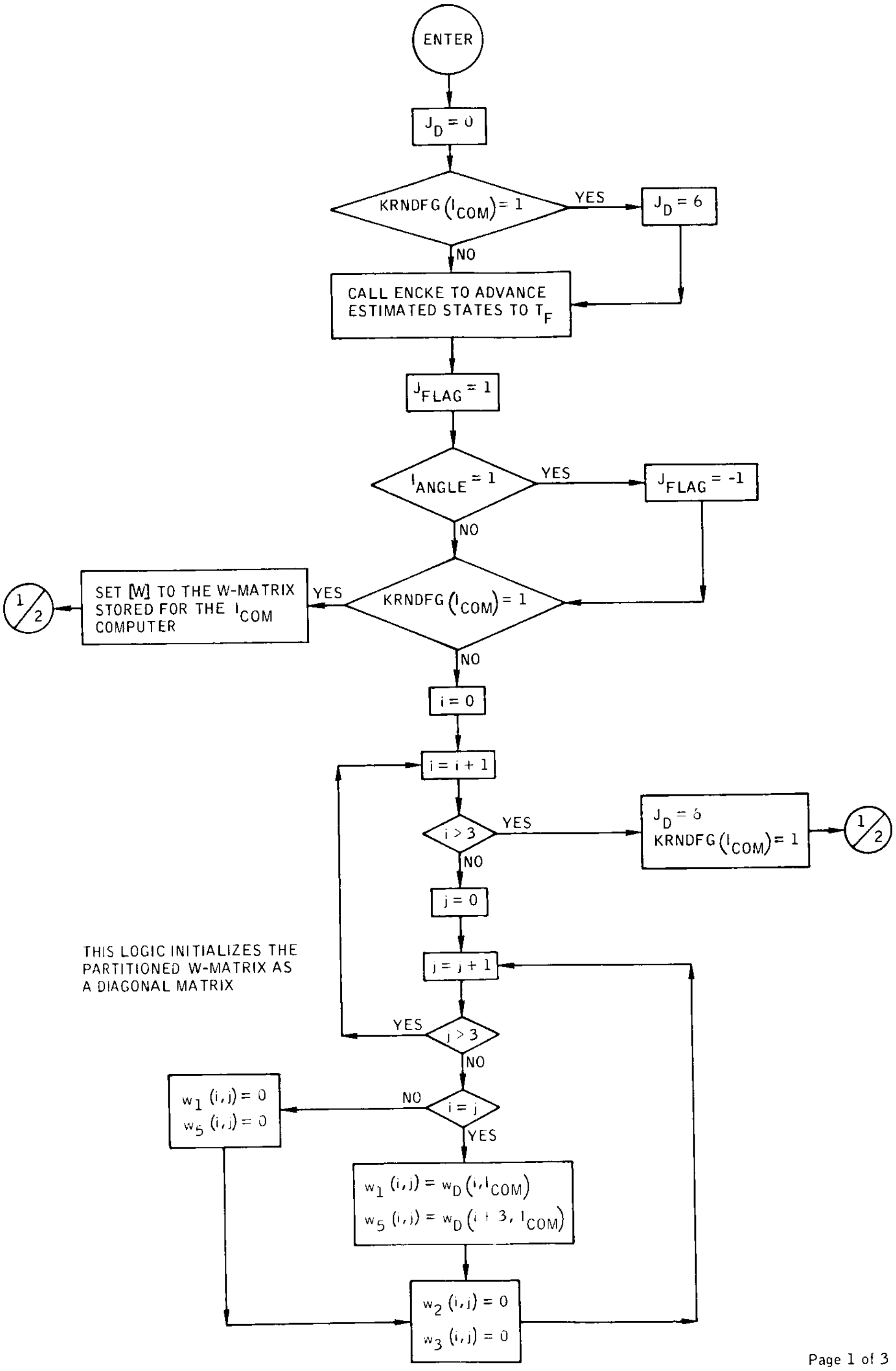

15
BEGIN LOOP TO TURN ON MONITORING FLAGS FOR COMPUTERS INTO WHICH VECTORS WERE LOADED
$I_{PASS} = 1$
IS COMPUTER I_{PASS} ALREADY BEING MONITORED
$[I_{MON}(I_{PASS}) = 1]$
YES
NO
WERE VECTORS LOADED INTO COMPUTER I_{PASS}
$[I_{UPDATE}(I_{PASS}) = 0]$
NO
YES
$I_{MON}(I_{PASS}) = 1$
RETURN

Figure B-74.- Concluded.



Page 1 of 3

Figure B-75.- Flow chart of subroutine RNR.

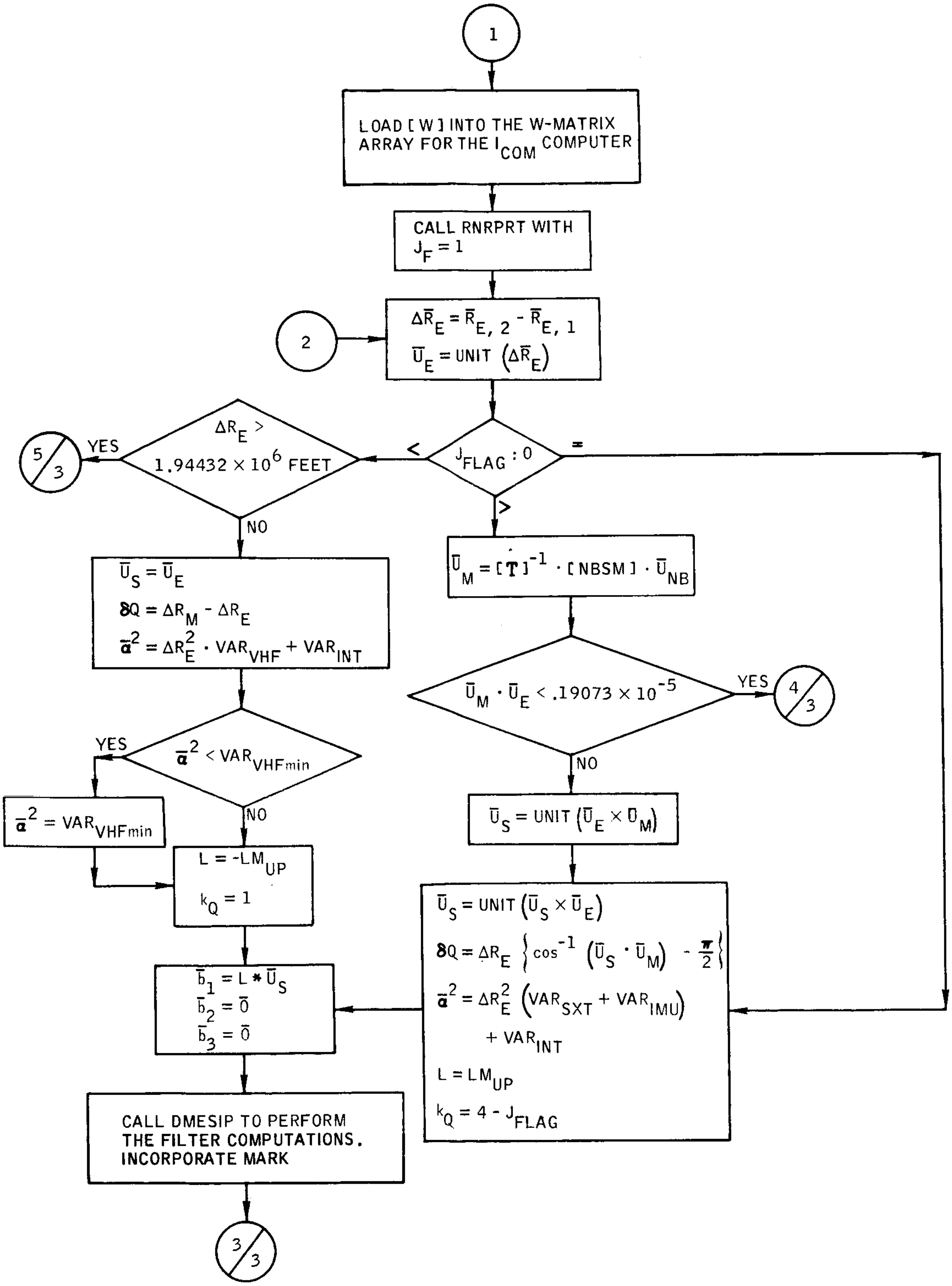
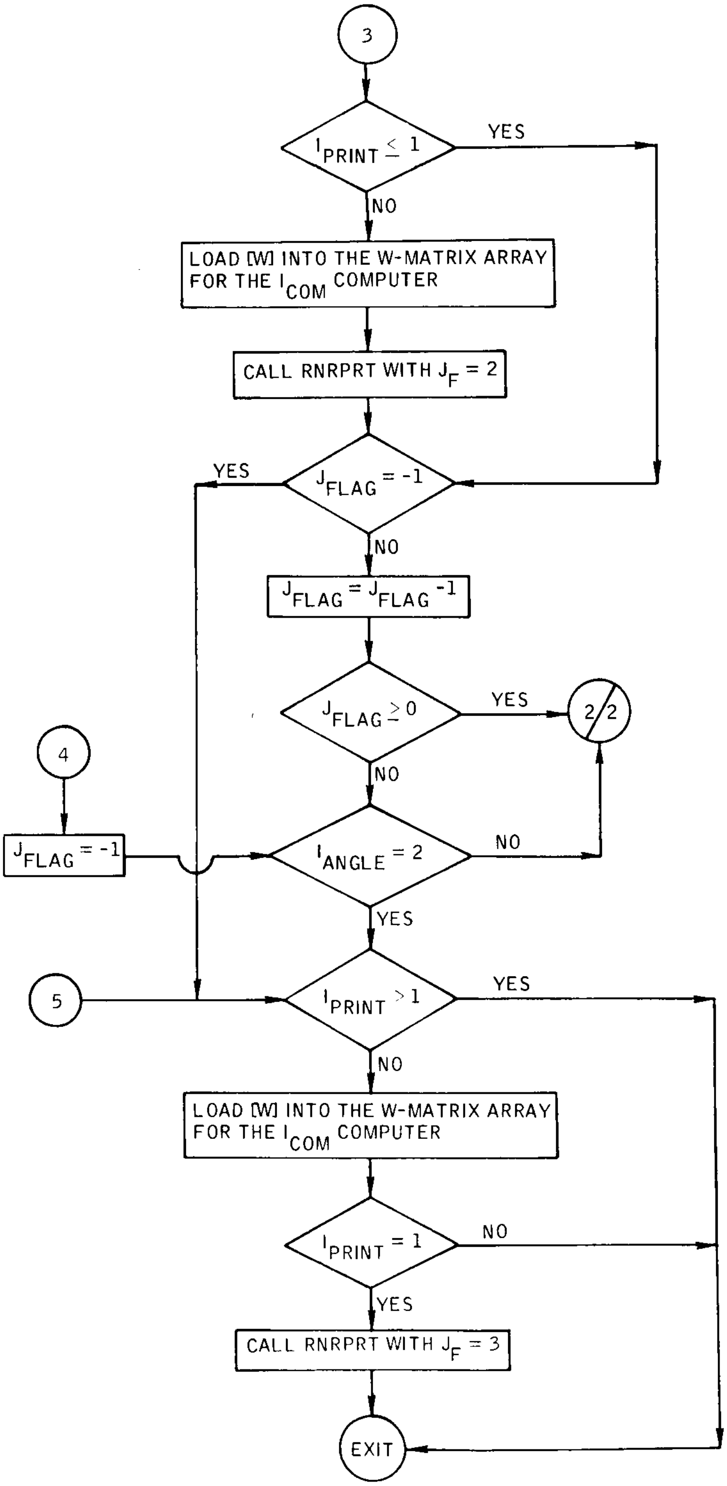

1
LOAD [W] INTO THE W-MATRIX ARRAY FOR THE I_{COM} COMPUTER
CALL RNRPRT WITH $J_F = 1$
2
$\Delta\bar{R}_E = \bar{R}_{E,2} - \bar{R}_{E,1}$
$\bar{U}_E = \text{UNIT}\left(\Delta\bar{R}_E\right)$
$J_{FLAG} : 0$
<
=
>
$\Delta R_E > 1.94432 \times 10^6$ FEET
YES
5/3
NO
$\bar{U}_S = \bar{U}_E$
$\delta Q = \Delta R_M - \Delta R_E$
$\bar{\alpha}^2 = \Delta R_E^2 \cdot VAR_{VHF} + VAR_{INT}$
$\bar{\alpha}^2 < VAR_{VHFmin}$
YES
$\bar{\alpha}^2 = VAR_{VHFmin}$
NO
$L = -LM_{UP}$
$k_Q = 1$
$\bar{b}_1 = L * \bar{U}_S$
$\bar{b}_2 = \bar{0}$
$\bar{b}_3 = \bar{0}$
CALL DMESIP TO PERFORM THE FILTER COMPUTATIONS. INCORPORATE MARK
3/3
$\bar{U}_M = [\dot{T}]^{-1} \cdot [NBSM] \cdot \bar{U}_{NB}$
$\bar{U}_M \cdot \bar{U}_E < .19073 \times 10^{-5}$
YES
4/3
NO
$\bar{U}_S = \text{UNIT}\left(\bar{U}_E \times \bar{U}_M\right)$
$\bar{U}_S = \text{UNIT}\left(\bar{U}_S \times \bar{U}_E\right)$
$\delta Q = \Delta R_E \left\{\cos^{-1}\left(\bar{U}_S \cdot \bar{U}_M\right) - \frac{\pi}{2}\right\}$
$\bar{\alpha}^2 = \Delta R_E^2 \left(VAR_{SXT} + VAR_{IMU}\right) + VAR_{INT}$
$L = LM_{UP}$
$k_Q = 4 - J_{FLAG}$

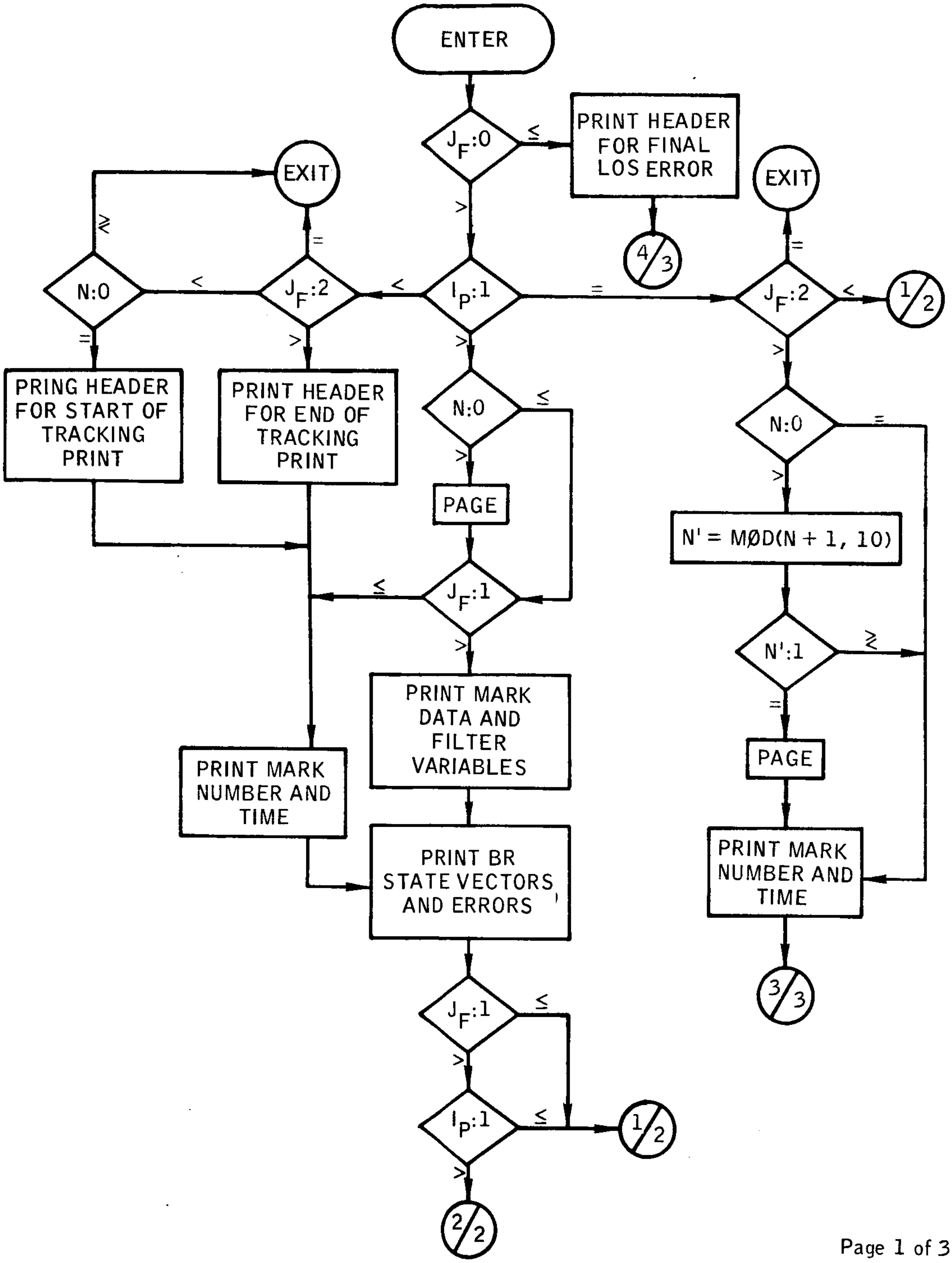
Figure B-75.- Continued.

Page 2 of 3


3
$I_{PRINT} \leq 1$
YES
NO
LOAD [W] INTO THE W-MATRIX ARRAY FOR THE I_{COM} COMPUTER
CALL RNRPRT WITH $J_F = 2$
YES
$J_{FLAG} = -1$
NO
$J_{FLAG} = J_{FLAG} - 1$
$J_{FLAG} \geq 0$
YES
2/2
NO
4
$J_{FLAG} = -1$
$I_{ANGLE} = 2$
NO
YES
5
$I_{PRINT} > 1$
YES
NO
LOAD [W] INTO THE W-MATRIX ARRAY FOR THE I_{COM} COMPUTER
$I_{PRINT} = 1$
NO
YES
CALL RNRPRT WITH $J_F = 3$
EXIT

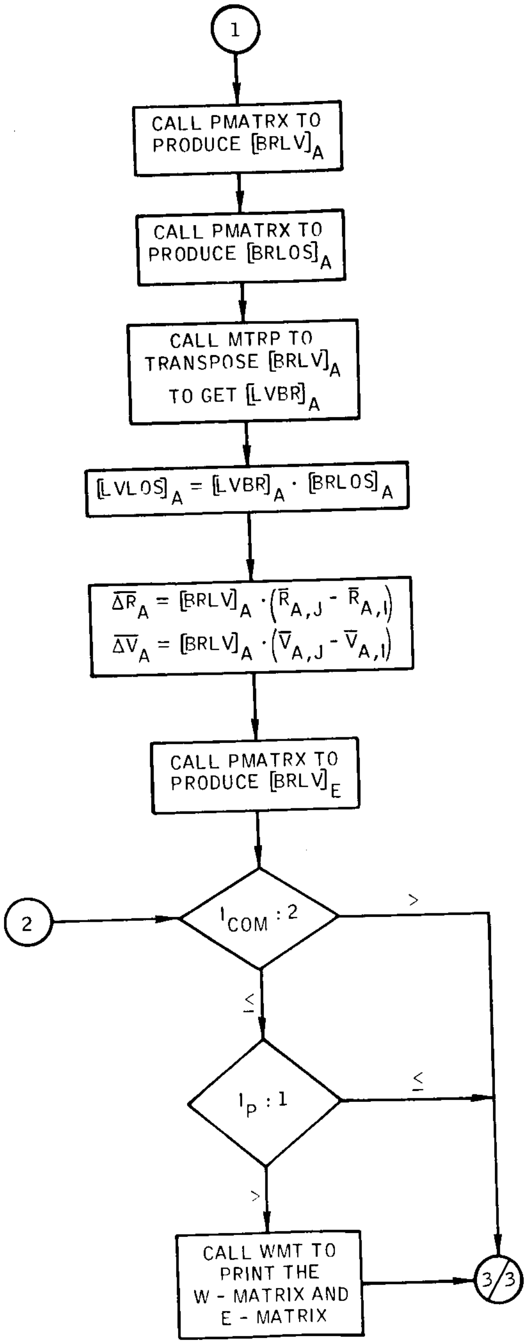
Page 3 of 3

Figure B-75.- Concluded.


ENTER
J_F:0
≤
PRINT HEADER FOR FINAL LOS ERROR
4/3
>
EXIT
EXIT
≷
=
=
N:0
<
J_F:2
<
I_P:1
=
J_F:2
<
1/2
=
>
>
>
PRING HEADER FOR START OF TRACKING PRINT
PRINT HEADER FOR END OF TRACKING PRINT
N:0
≤
N:0
=
>
>
PAGE
N' = MØD(N + 1, 10)
≤
J_F:1
>
N':1
≷
=
PRINT MARK DATA AND FILTER VARIABLES
PAGE
PRINT MARK NUMBER AND TIME
PRINT BR STATE VECTORS AND ERRORS
PRINT MARK NUMBER AND TIME
3/3
J_F:1
≤
>
I_P:1
≤
1/2
>
2/2

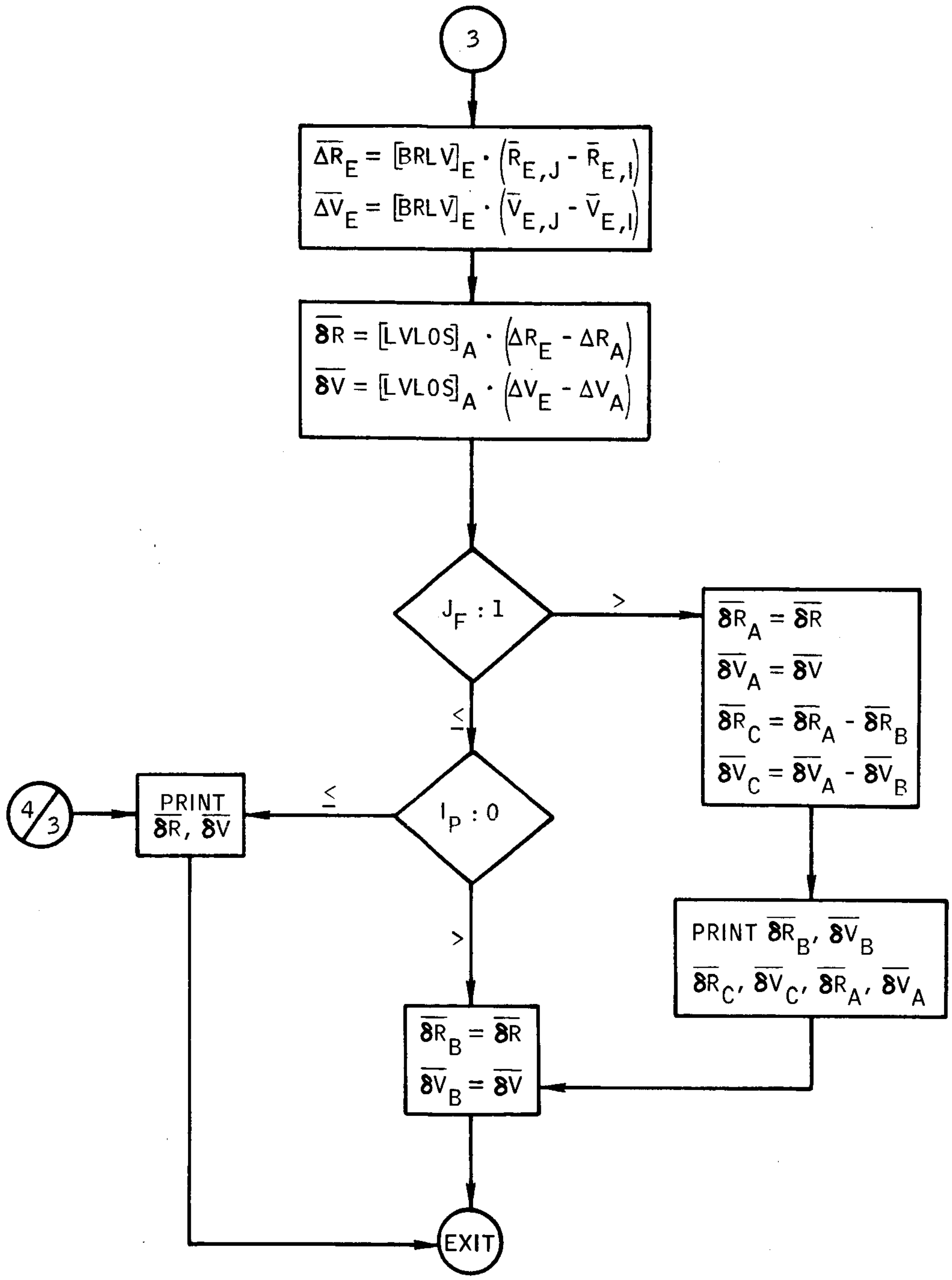
Page 1 of 3

Figure B-76.- Flow chart of subroutine RNRPRT.


1
CALL PMATRX TO PRODUCE $[BRLV]_A$
CALL PMATRX TO PRODUCE $[BRLOS]_A$
CALL MTRP TO TRANSPOSE $[BRLV]_A$ TO GET $[LVBR]_A$
$[LVLOS]_A = [LVBR]_A \cdot [BRLOS]_A$
$\overline{\Delta R}_A = [BRLV]_A \cdot (\overline{R}_{A,J} - \overline{R}_{A,I})$
$\overline{\Delta V}_A = [BRLV]_A \cdot (\overline{V}_{A,J} - \overline{V}_{A,I})$
CALL PMATRX TO PRODUCE $[BRLV]_E$
2
$I_{COM} : 2$
>
≤
$I_P : 1$
≤
>
CALL WMT TO PRINT THE W - MATRIX AND E - MATRIX
3/3

Page 2 of 3

Figure B-76.- Continued.


3
$\overline{\Delta R}_E = [BRLV]_E \cdot (\bar{R}_{E,J} - \bar{R}_{E,I})$
$\overline{\Delta V}_E = [BRLV]_E \cdot (\bar{V}_{E,J} - \bar{V}_{E,I})$
$\overline{\delta R} = [LVLOS]_A \cdot (\Delta R_E - \Delta R_A)$
$\overline{\delta V} = [LVLOS]_A \cdot (\Delta V_E - \Delta V_A)$
$J_F : 1$
>
$\overline{\delta R}_A = \overline{\delta R}$
$\overline{\delta V}_A = \overline{\delta V}$
$\overline{\delta R}_C = \overline{\delta R}_A - \overline{\delta R}_B$
$\overline{\delta V}_C = \overline{\delta V}_A - \overline{\delta V}_B$
≤
$I_P : 0$
≤
PRINT $\overline{\delta R}$, $\overline{\delta V}$
4/3
>
PRINT $\overline{\delta R}_B$, $\overline{\delta V}_B$
$\overline{\delta R}_C$, $\overline{\delta V}_C$, $\overline{\delta R}_A$, $\overline{\delta V}_A$
$\overline{\delta R}_B = \overline{\delta R}$
$\overline{\delta V}_B = \overline{\delta V}$
EXIT

Page 3 of 3

Figure B-76.- Concluded.

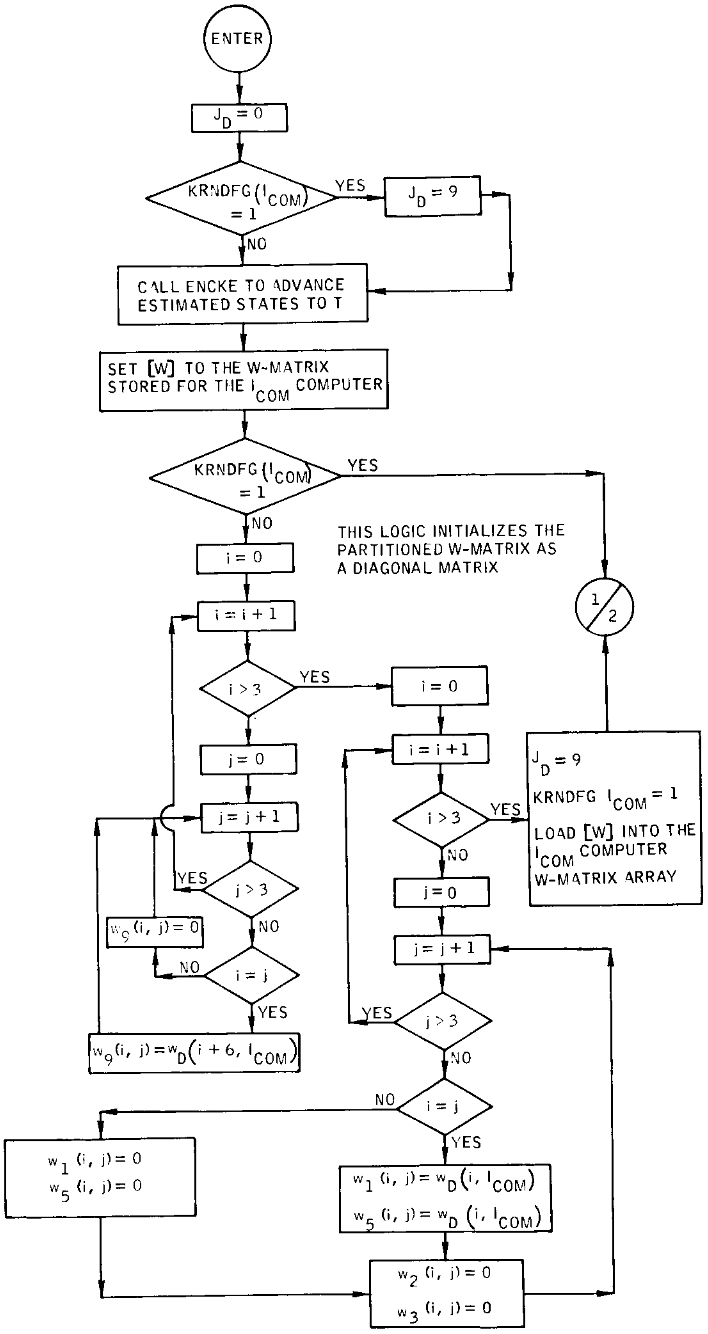

ENTER
$J_D = 0$
KRNDFG (I_{COM}) = 1
YES
$J_D = 9$
NO
CALL ENCKE TO ADVANCE ESTIMATED STATES TO T
SET [W] TO THE W-MATRIX STORED FOR THE I_{COM} COMPUTER
KRNDFG (I_{COM}) = 1
YES
NO
THIS LOGIC INITIALIZES THE PARTITIONED W-MATRIX AS A DIAGONAL MATRIX
i = 0
i = i + 1
i > 3
YES
i = 0
j = 0
j = j + 1
YES
j > 3
NO
w_9 (i, j) = 0
NO
i = j
YES
w_9(i, j) = w_D(i + 6, I_{COM})
i = i + 1
i > 3
YES
NO
j = 0
j = j + 1
YES
j > 3
NO
NO
i = j
YES
w_1 (i, j) = 0
w_5 (i, j) = 0
w_1 (i, j) = w_D(i, I_{COM})
w_5 (i, j) = w_D (i, I_{COM})
w_2 (i, j) = 0
w_3 (i, j) = 0
$J_D = 9$
KRNDFG I_{COM} = 1
LOAD [W] INTO THE I_{COM} COMPUTER W-MATRIX ARRAY
1/2

Figure B-77.- Flow chart of subroutine RR.

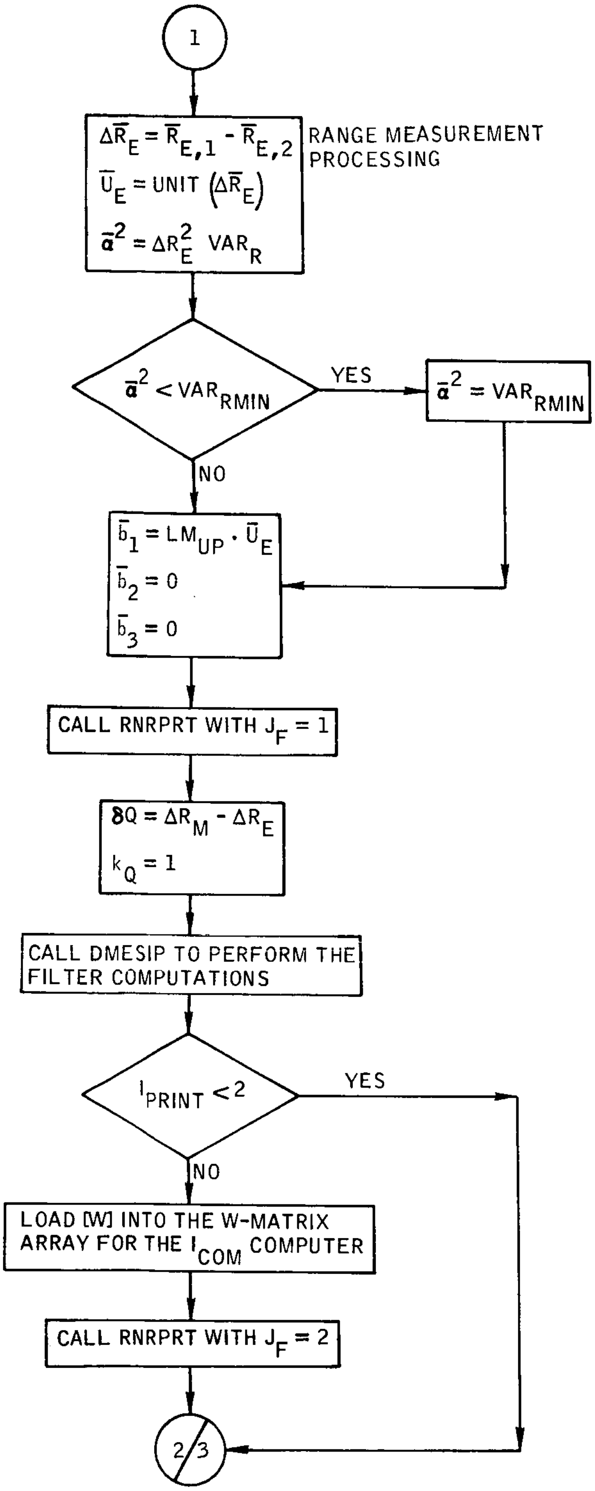

1
$\Delta\bar{R}_E = \bar{R}_{E,1} - \bar{R}_{E,2}$
$\bar{U}_E = \text{UNIT}\left(\Delta\bar{R}_E\right)$
$\bar{\alpha}^2 = \Delta R_E^2 \text{ VAR}_R$
RANGE MEASUREMENT PROCESSING
$\bar{\alpha}^2 < \text{VAR}_{\text{RMIN}}$
YES
$\bar{\alpha}^2 = \text{VAR}_{\text{RMIN}}$
NO
$\bar{b}_1 = \text{LM}_{\text{UP}} \cdot \bar{U}_E$
$\bar{b}_2 = 0$
$\bar{b}_3 = 0$
CALL RNRPRT WITH $J_F = 1$
$\delta Q = \Delta R_M - \Delta R_E$
$k_Q = 1$
CALL DMESIP TO PERFORM THE FILTER COMPUTATIONS
$I_{\text{PRINT}} < 2$
YES
NO
LOAD [W] INTO THE W-MATRIX ARRAY FOR THE I_{COM} COMPUTER
CALL RNRPRT WITH $J_F = 2$
2/3

Figure B-77.- Continued.

Page 2 of 5

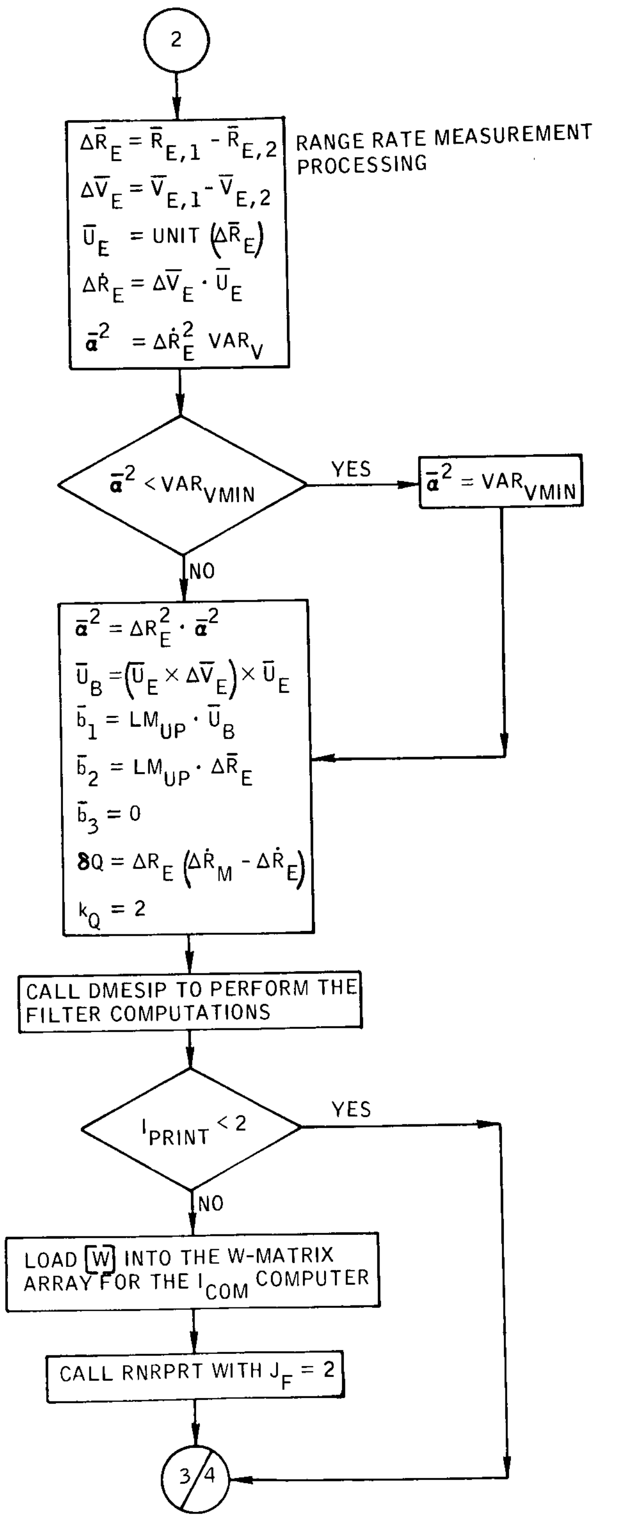

2
$\Delta\bar{R}_E = \bar{R}_{E,1} - \bar{R}_{E,2}$
$\Delta\bar{V}_E = \bar{V}_{E,1} - \bar{V}_{E,2}$
$\bar{U}_E = \text{UNIT}\left(\Delta\bar{R}_E\right)$
$\Delta\dot{R}_E = \Delta\bar{V}_E \cdot \bar{U}_E$
$\bar{\alpha}^2 = \Delta\dot{R}_E^2\ \text{VAR}_V$
RANGE RATE MEASUREMENT PROCESSING
$\bar{\alpha}^2 < \text{VAR}_{\text{VMIN}}$
YES
$\bar{\alpha}^2 = \text{VAR}_{\text{VMIN}}$
NO
$\bar{\alpha}^2 = \Delta R_E^2 \cdot \bar{\alpha}^2$
$\bar{U}_B = \left(\bar{U}_E \times \Delta\bar{V}_E\right) \times \bar{U}_E$
$\bar{b}_1 = LM_{UP} \cdot \bar{U}_B$
$\bar{b}_2 = LM_{UP} \cdot \Delta\bar{R}_E$
$\bar{b}_3 = 0$
$\delta Q = \Delta R_E \left(\Delta\dot{R}_M - \Delta\dot{R}_E\right)$
$k_Q = 2$
CALL DMESIP TO PERFORM THE FILTER COMPUTATIONS
$I_{PRINT} < 2$
YES
NO
LOAD [W] INTO THE W-MATRIX ARRAY FOR THE I_{COM} COMPUTER
CALL RNRPRT WITH $J_F = 2$
3/4

Figure B-77.- Continued.

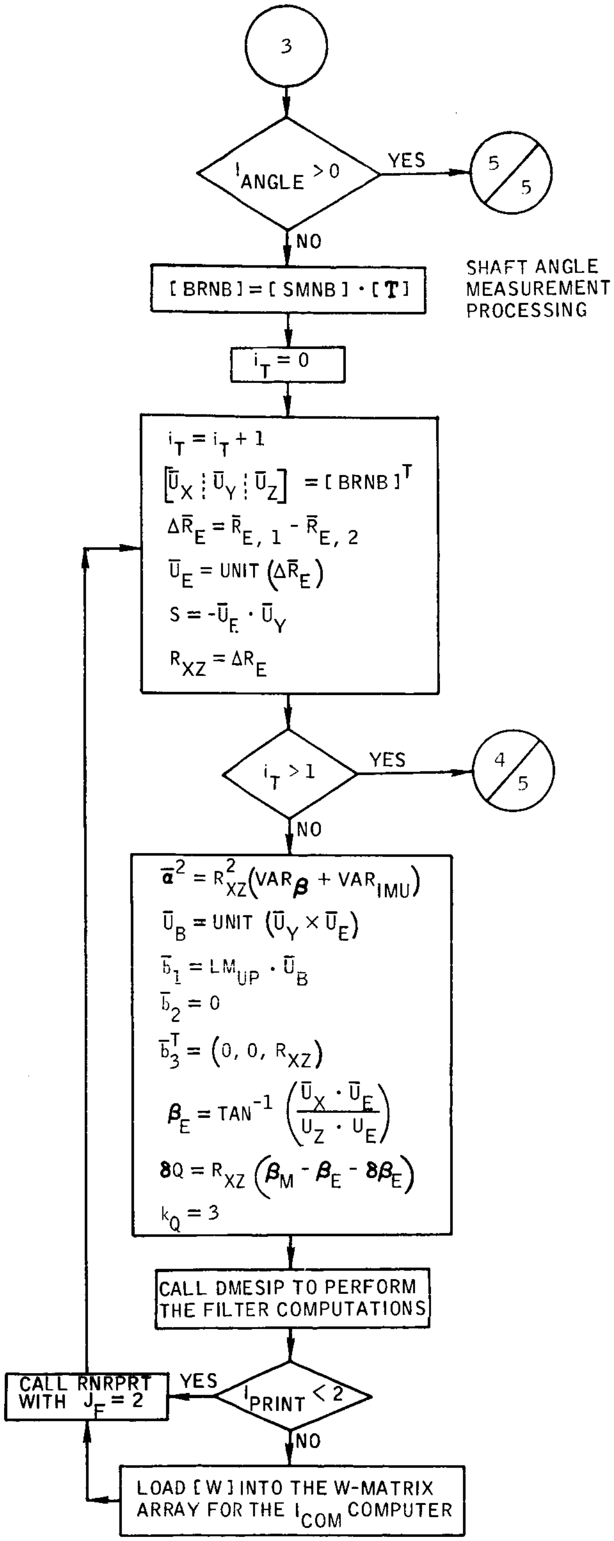

3
$i_{ANGLE} > 0$
YES
5/5
NO
$[BRNB] = [SMNB] \cdot [T]$
SHAFT ANGLE MEASUREMENT PROCESSING
$i_T = 0$
$i_T = i_T + 1$
$[\bar{U}_X \vdots \bar{U}_Y \vdots \bar{U}_Z] = [BRNB]^T$
$\Delta\bar{R}_E = \bar{R}_{E,1} - \bar{R}_{E,2}$
$\bar{U}_E = UNIT(\Delta\bar{R}_E)$
$S = -\bar{U}_E \cdot \bar{U}_Y$
$R_{XZ} = \Delta R_E$
$i_T > 1$
YES
4/5
NO
$\bar{\alpha}^2 = R_{XZ}^2(VAR_{\beta} + VAR_{IMU})$
$\bar{U}_B = UNIT(\bar{U}_Y \times \bar{U}_E)$
$\bar{b}_1 = LM_{UP} \cdot \bar{U}_B$
$\bar{b}_2 = 0$
$\bar{b}_3^T = (0, 0, R_{XZ})$
$\beta_E = TAN^{-1}\left(\frac{\bar{U}_X \cdot \bar{U}_E}{U_Z \cdot U_E}\right)$
$\delta Q = R_{XZ}(\beta_M - \beta_E - \delta\beta_E)$
$k_Q = 3$
CALL DMESIP TO PERFORM THE FILTER COMPUTATIONS
$i_{PRINT} < 2$
YES
CALL RNRPRT WITH $J_F = 2$
NO
LOAD [W] INTO THE W-MATRIX ARRAY FOR THE I_{COM} COMPUTER

Figure B-77.- Continued.

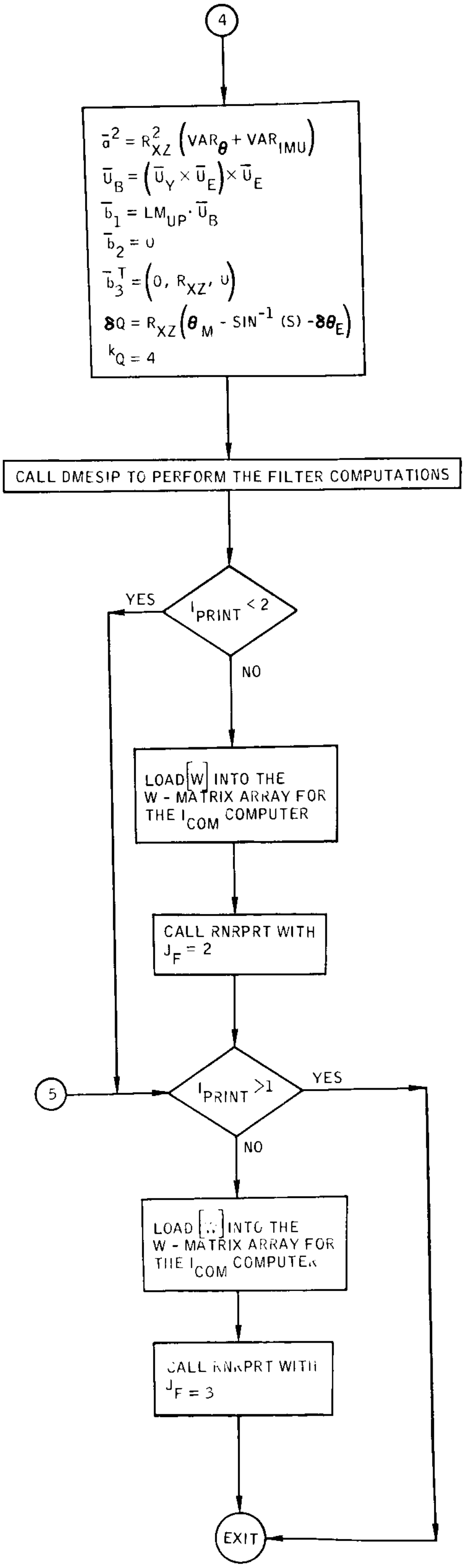

4
$\bar{a}^2 = R_{XZ}^2 \left(VAR_\theta + VAR_{IMU}\right)$
$\bar{U}_B = \left(\bar{U}_Y \times \bar{U}_E\right) \times \bar{U}_E$
$\bar{b}_1 = LM_{UP} \cdot \bar{U}_B$
$\bar{b}_2 = 0$
$\bar{b}_3^T = \left(0, R_{XZ}, 0\right)$
$\delta Q = R_{XZ}\left(\theta_M - SIN^{-1}(S) - \delta\theta_E\right)$
$k_Q = 4$
CALL DMESIP TO PERFORM THE FILTER COMPUTATIONS
$I_{PRINT} < 2$
YES
NO
LOAD [W] INTO THE W - MATRIX ARRAY FOR THE I_{COM} COMPUTER
CALL RNRPRT WITH $J_F = 2$
5
$I_{PRINT} > 1$
YES
NO
LOAD [W] INTO THE W - MATRIX ARRAY FOR THE I_{COM} COMPUTER
CALL RNRPRT WITH $J_F = 3$
EXIT

Figure B-77.- Concluded.

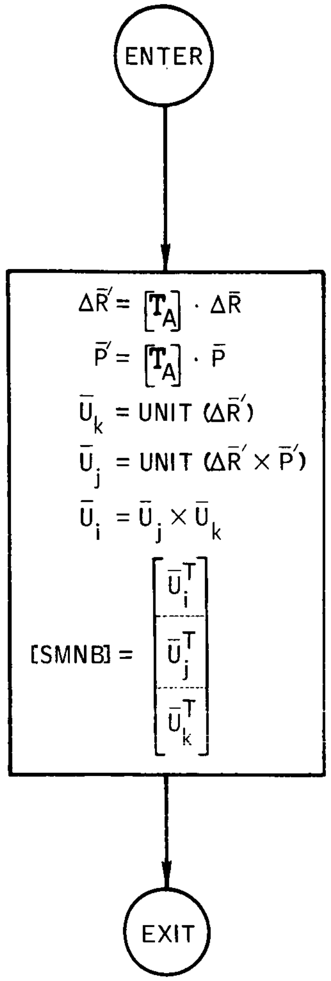
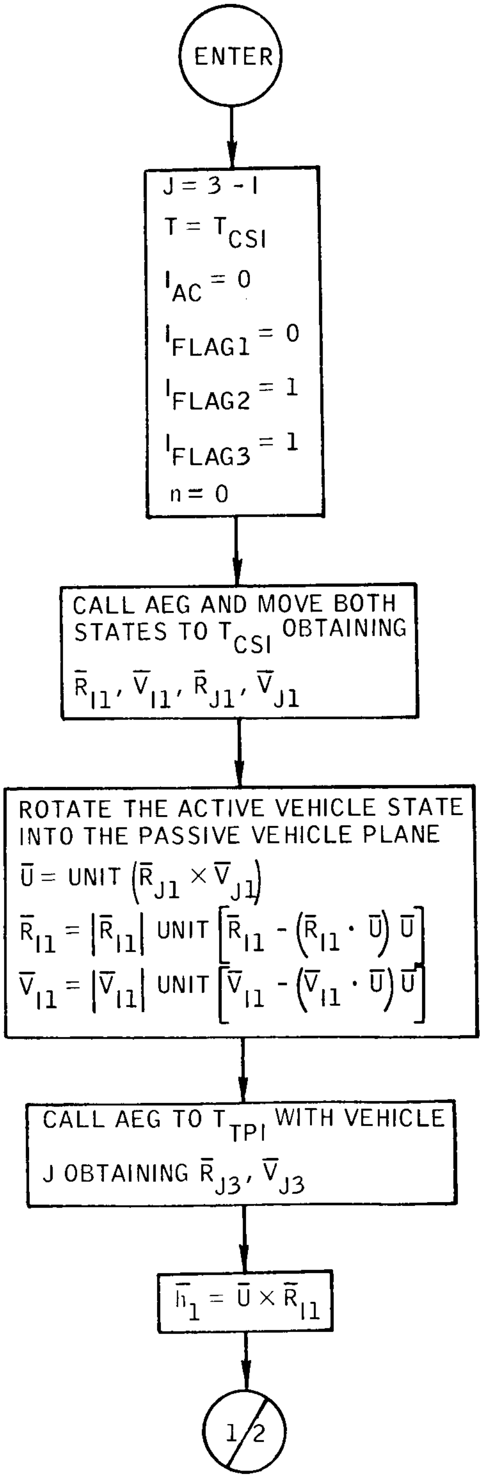

ENTER
$\Delta\bar{R}' = [T_A] \cdot \Delta\bar{R}$
$\bar{P}' = [T_A] \cdot \bar{P}$
$\bar{U}_k = \text{UNIT}(\Delta\bar{R}')$
$\bar{U}_j = \text{UNIT}(\Delta\bar{R}' \times \bar{P}')$
$\bar{U}_i = \bar{U}_j \times \bar{U}_k$
$[\text{SMNB}] = \begin{bmatrix} \bar{U}_i^T \\ \hdashline \bar{U}_j^T \\ \hdashline \bar{U}_k^T \end{bmatrix}$
EXIT

Figure B-78.- Flow chart of subroutine PRSMNB.

Page 1 of 1


ENTER
$J = 3 - I$
$T = T_{CSI}$
$I_{AC} = 0$
$I_{FLAG1} = 0$
$I_{FLAG2} = 1$
$I_{FLAG3} = 1$
$n = 0$
CALL AEG AND MOVE BOTH STATES TO T_{CSI} OBTAINING
$\bar{R}_{I1}, \bar{V}_{I1}, \bar{R}_{J1}, \bar{V}_{J1}$
ROTATE THE ACTIVE VEHICLE STATE INTO THE PASSIVE VEHICLE PLANE
$\bar{U} = \text{UNIT}\left(\bar{R}_{J1} \times \bar{V}_{J1}\right)$
$\bar{R}_{I1} = \left|\bar{R}_{I1}\right| \text{UNIT}\left[\bar{R}_{I1} - \left(\bar{R}_{I1} \cdot \bar{U}\right)\bar{U}\right]$
$\bar{V}_{I1} = \left|\bar{V}_{I1}\right| \text{UNIT}\left[\bar{V}_{I1} - \left(\bar{V}_{I1} \cdot \bar{U}\right)\bar{U}\right]$
CALL AEG TO T_{TPI} WITH VEHICLE J OBTAINING $\bar{R}_{J3}, \bar{V}_{J3}$
$\bar{h}_1 = \bar{U} \times \bar{R}_{I1}$
1/2

Page 1 of 7

Figure B-79.- Flow chart of subroutine RTCCSI.

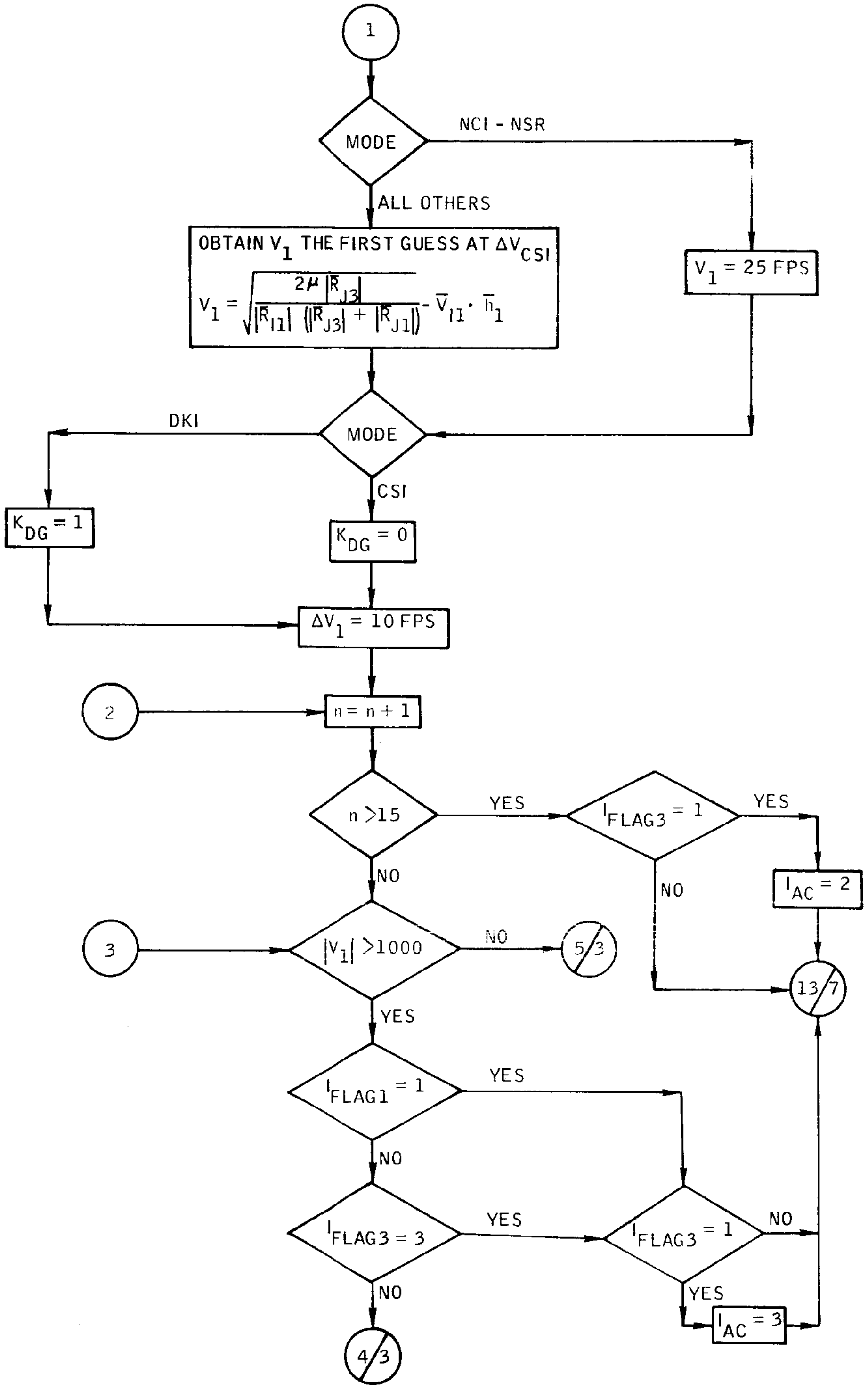
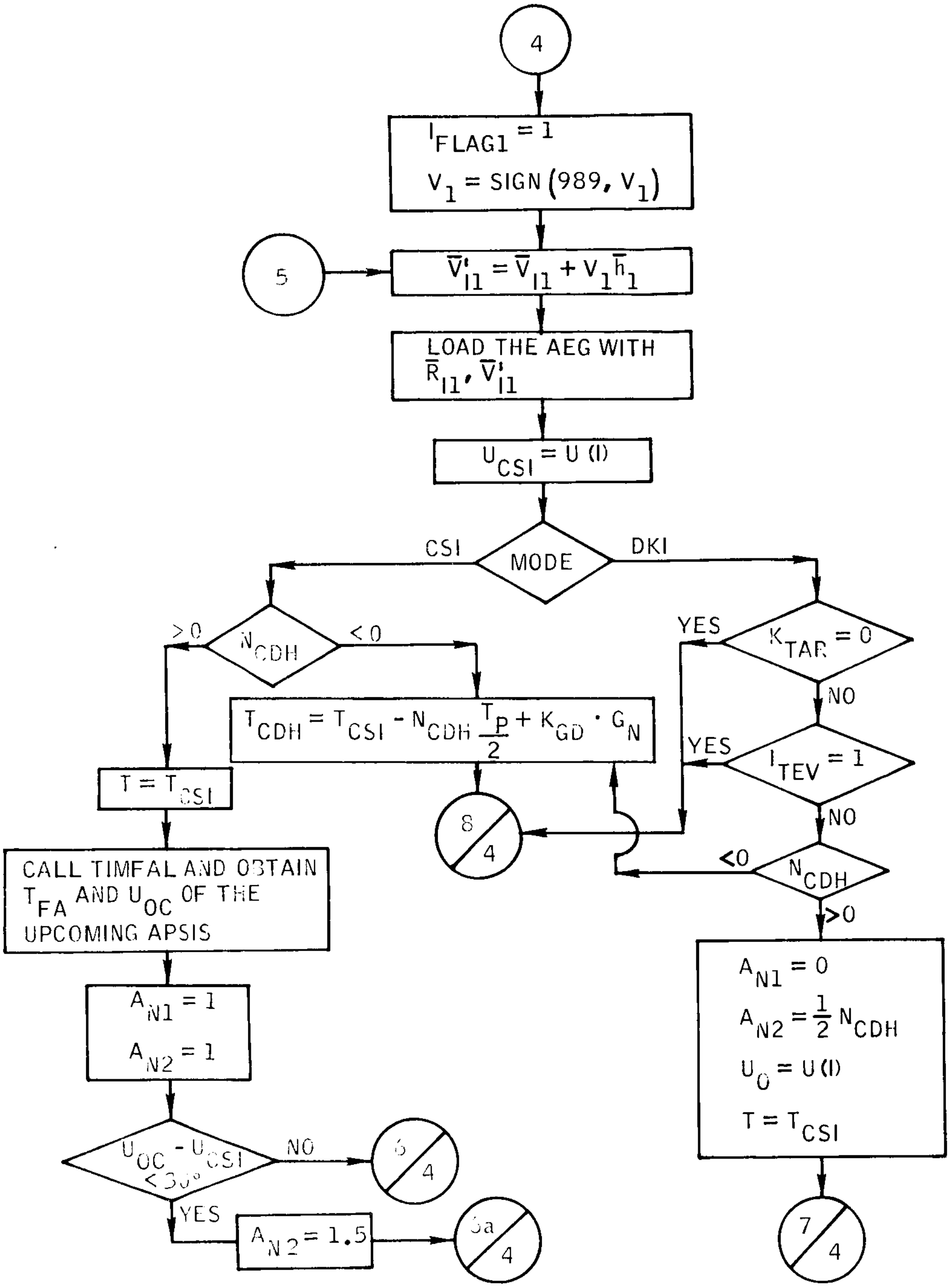


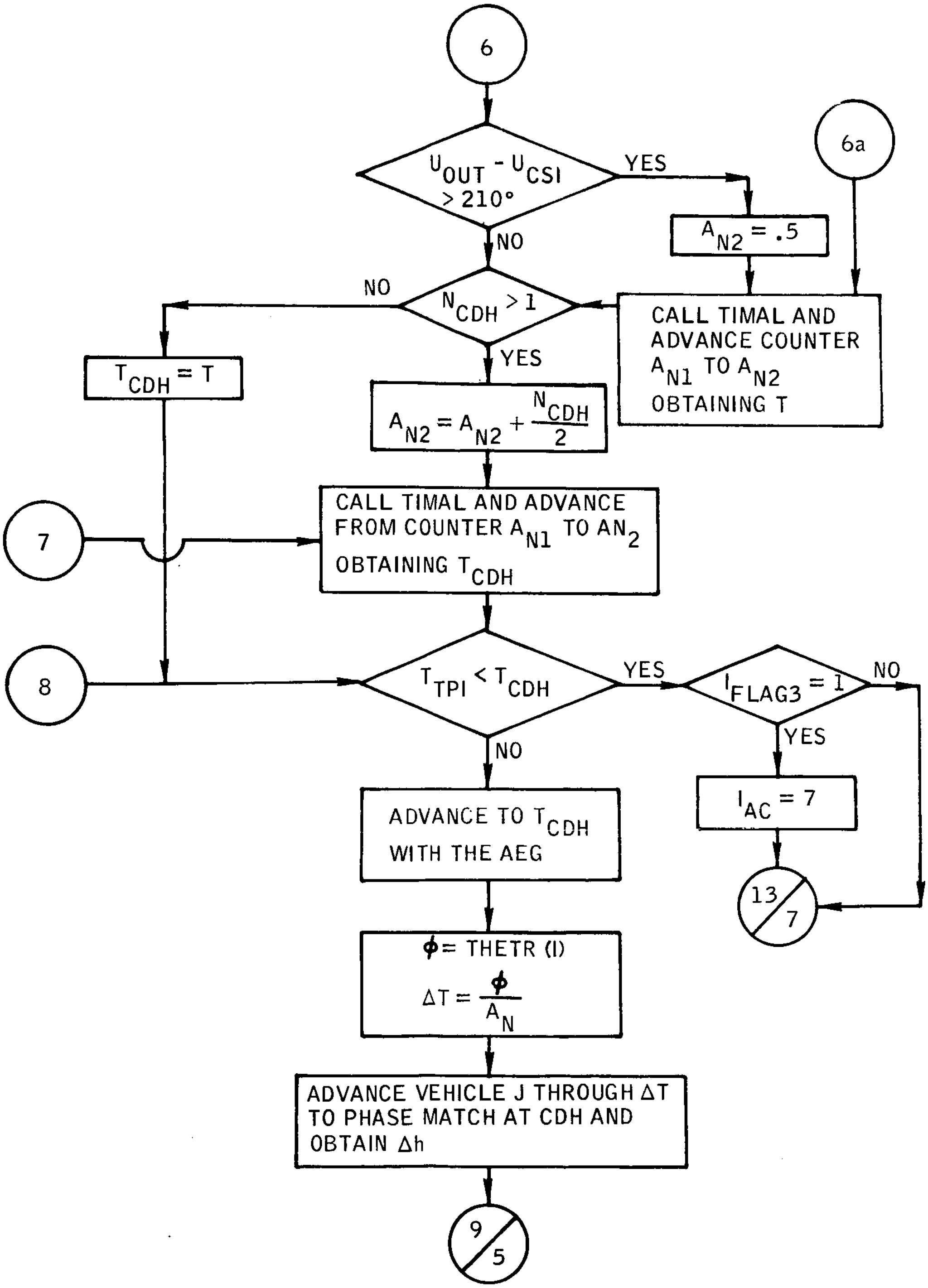
Figure B-79.- Continued.

Page 2 of 7



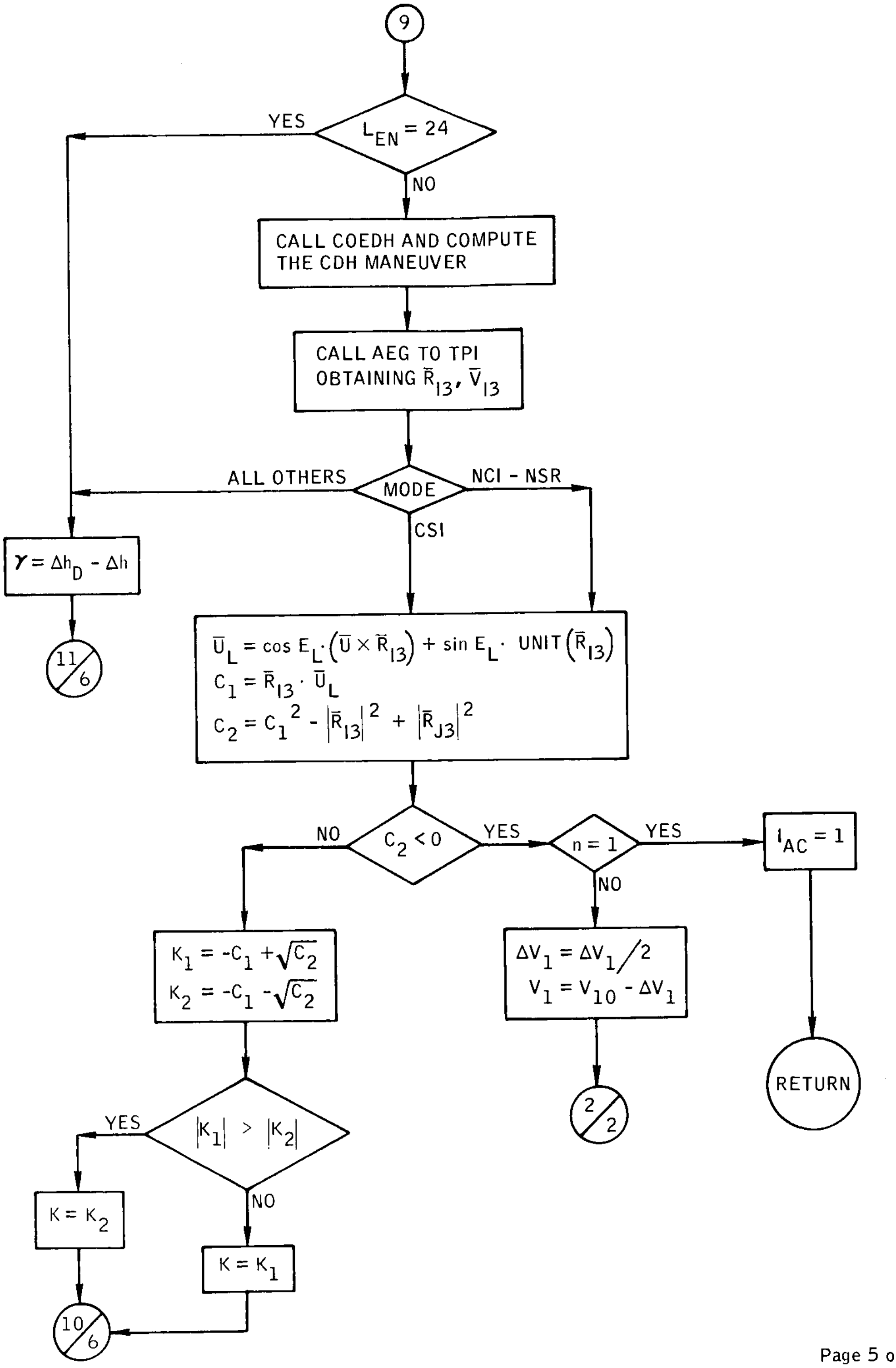
Page 3 of 7

Figure B-79.- Continued.


6
$U_{OUT} - U_{CSI} > 210°$
YES
NO
6a
$A_{N2} = .5$
$N_{CDH} > 1$
NO
YES
CALL TIMAL AND ADVANCE COUNTER A_{N1} TO A_{N2} OBTAINING T
$T_{CDH} = T$
$A_{N2} = A_{N2} + \frac{N_{CDH}}{2}$
7
CALL TIMAL AND ADVANCE FROM COUNTER A_{N1} TO AN_2 OBTAINING T_{CDH}
8
$T_{TPI} < T_{CDH}$
YES
$I_{FLAG3} = 1$
NO
YES
NO
$I_{AC} = 7$
13/7
ADVANCE TO T_{CDH} WITH THE AEG
ϕ = THETR (I)
$\Delta T = \frac{\phi}{A_N}$
ADVANCE VEHICLE J THROUGH ΔT TO PHASE MATCH AT CDH AND OBTAIN Δh
9/5

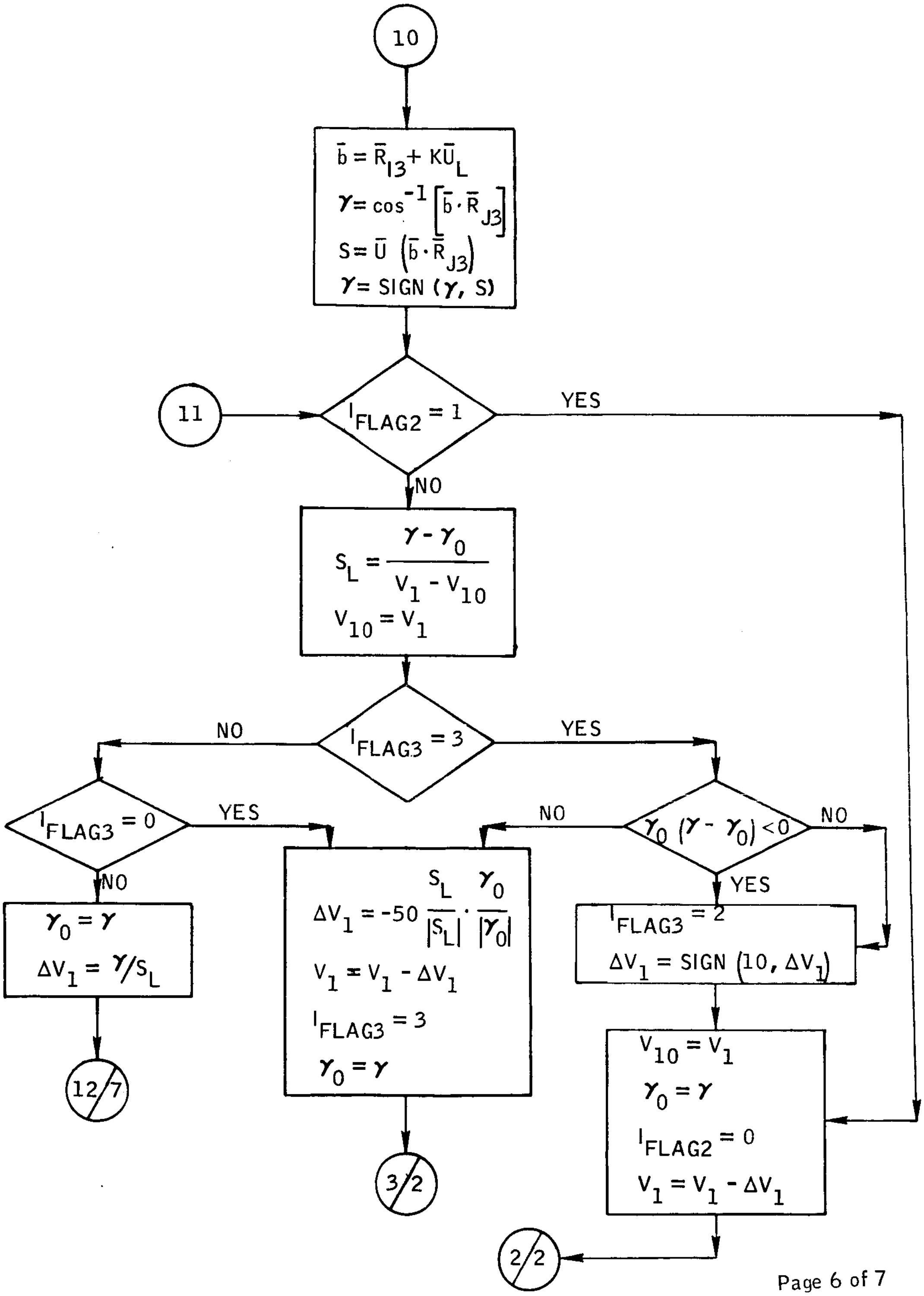
Page 4 of 7

Figure B-79.- Continued.


9
$L_{EN} = 24$
YES
NO
CALL COEDH AND COMPUTE THE CDH MANEUVER
CALL AEG TO TPI OBTAINING $\bar{R}_{13}, \bar{V}_{13}$
MODE
ALL OTHERS
NCI - NSR
CSI
$\gamma = \Delta h_D - \Delta h$
11/6
$\bar{U}_L = \cos E_L \cdot (\bar{U} \times \bar{R}_{13}) + \sin E_L \cdot \text{UNIT}(\bar{R}_{13})$
$C_1 = \bar{R}_{13} \cdot \bar{U}_L$
$C_2 = C_1{}^2 - |\bar{R}_{13}|^2 + |\bar{R}_{J3}|^2$
$C_2 < 0$
NO
YES
$n = 1$
YES
NO
$I_{AC} = 1$
RETURN
$\Delta V_1 = \Delta V_1 / 2$
$V_1 = V_{10} - \Delta V_1$
2/2
$K_1 = -C_1 + \sqrt{C_2}$
$K_2 = -C_1 - \sqrt{C_2}$
$|K_1| > |K_2|$
YES
NO
$K = K_2$
$K = K_1$
10/6

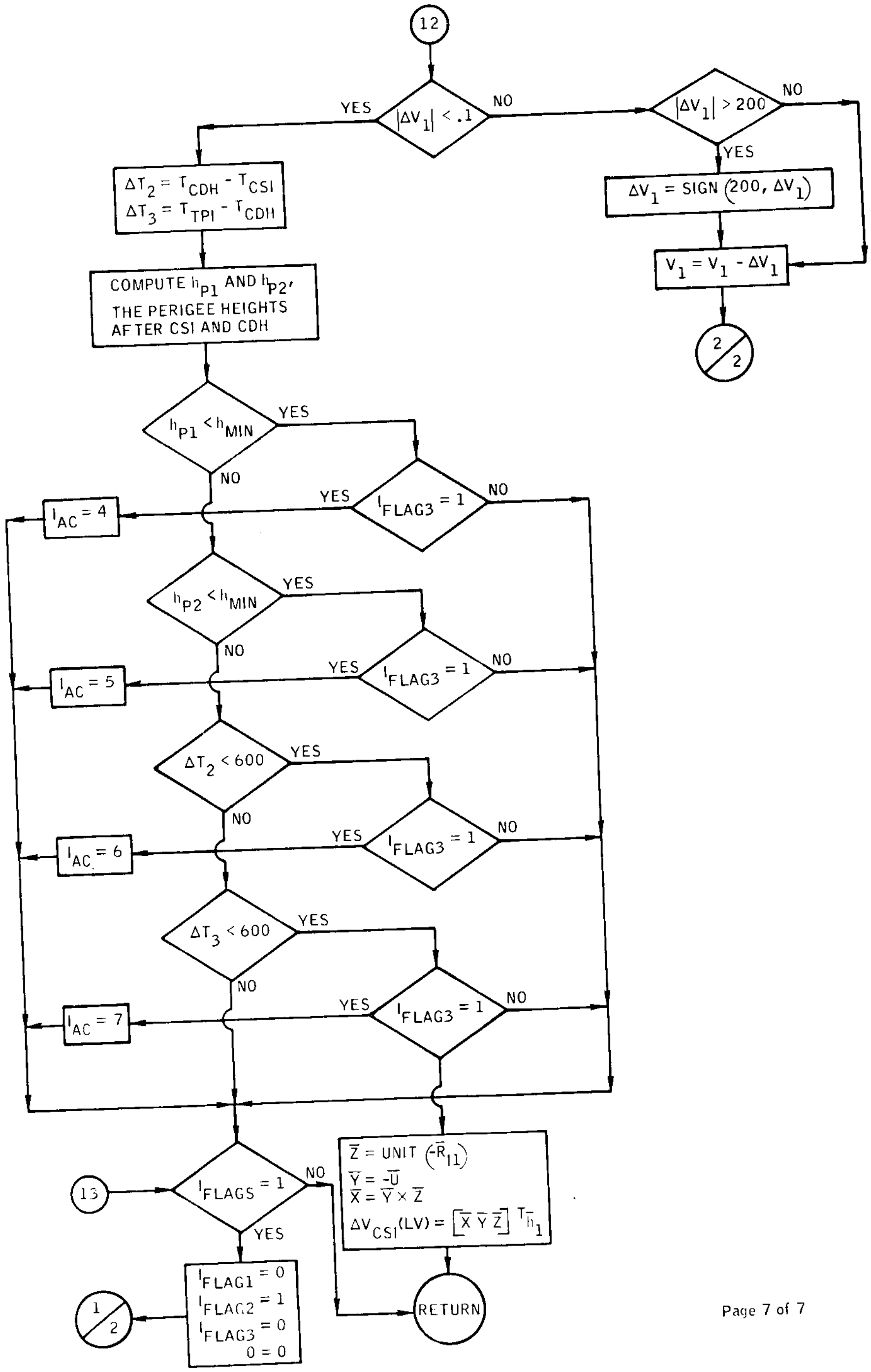
Page 5 of 7

Figure B-79.- Continued.


10
$\bar{b} = \bar{R}_{I3} + K\bar{U}_L$
$\gamma = \cos^{-1}\left[\bar{b} \cdot \bar{R}_{J3}\right]$
$S = \bar{U}\left(\bar{b} \cdot \bar{R}_{J3}\right)$
$\gamma = \text{SIGN}(\gamma, S)$
11
$I_{FLAG2} = 1$
YES
NO
$S_L = \frac{\gamma - \gamma_0}{V_1 - V_{10}}$
$V_{10} = V_1$
$I_{FLAG3} = 3$
NO
YES
$I_{FLAG3} = 0$
YES
NO
$\gamma_0 = \gamma$
$\Delta V_1 = \gamma / S_L$
12/7
$\Delta V_1 = -50 \frac{S_L}{|S_L|} \cdot \frac{\gamma_0}{|\gamma_0|}$
$V_1 = V_1 - \Delta V_1$
$I_{FLAG3} = 3$
$\gamma_0 = \gamma$
3/2
$\gamma_0 (\gamma - \gamma_0) < 0$
NO
NO
YES
$I_{FLAG3} = 2$
$\Delta V_1 = \text{SIGN}(10, \Delta V_1)$
$V_{10} = V_1$
$\gamma_0 = \gamma$
$I_{FLAG2} = 0$
$V_1 = V_1 - \Delta V_1$
2/2

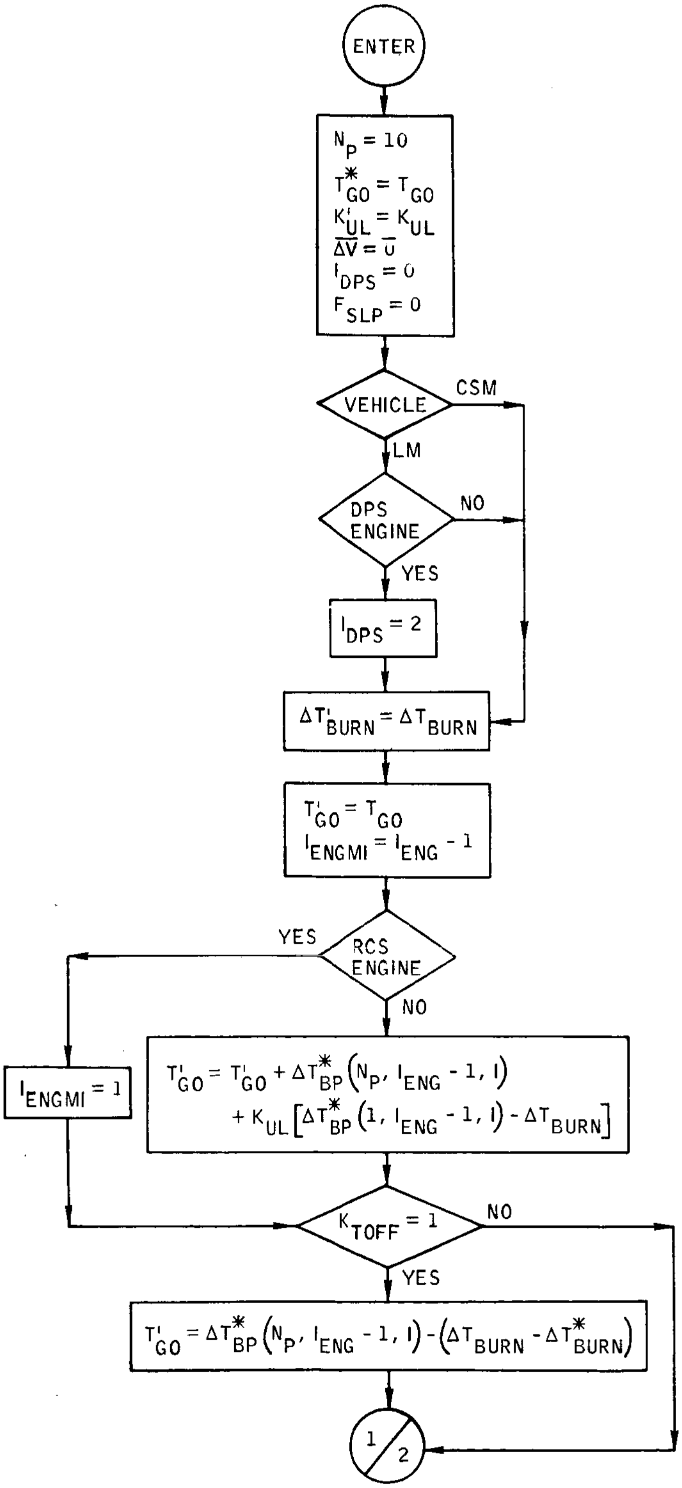
Page 6 of 7

Figure B-79.- Continued.



Page 7 of 7

Figure B-79.- Concluded.


ENTER
$N_P = 10$
$T^*_{GO} = T_{GO}$
$K'_{UL} = K_{UL}$
$\overline{\Delta V} = \overline{0}$
$I_{DPS} = 0$
$F_{SLP} = 0$
VEHICLE
CSM
LM
DPS ENGINE
NO
YES
$I_{DPS} = 2$
$\Delta T'_{BURN} = \Delta T_{BURN}$
$T'_{GO} = T_{GO}$
$I_{ENGMI} = I_{ENG} - 1$
RCS ENGINE
YES
NO
$I_{ENGMI} = 1$
$T'_{GO} = T'_{GO} + \Delta T^*_{BP}(N_P, I_{ENG} - 1, I) + K_{UL}\left[\Delta T^*_{BP}(1, I_{ENG} - 1, I) - \Delta T_{BURN}\right]$
$K_{TOFF} = 1$
NO
YES
$T'_{GO} = \Delta T^*_{BP}(N_P, I_{ENG} - 1, I) - (\Delta T_{BURN} - \Delta T^*_{BURN})$
1/2

Page 1 of 9

Figure B-80.- Flow chart of subroutine RUNGA.

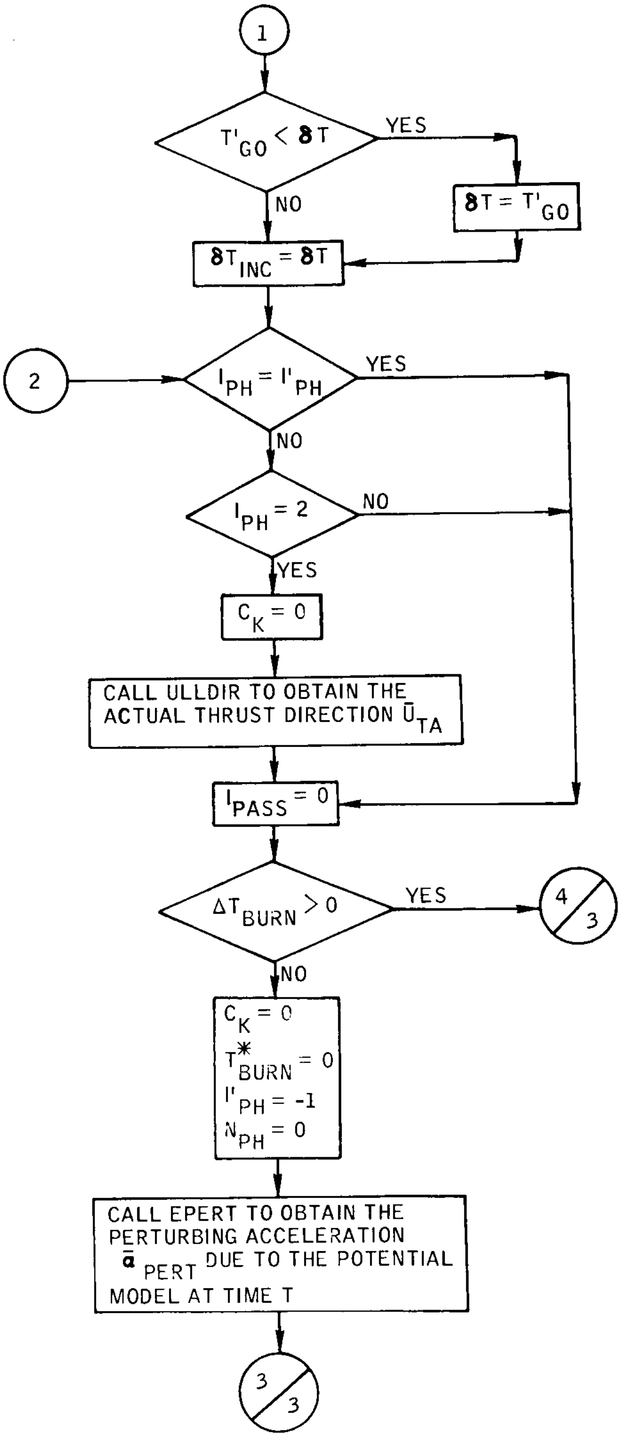

1
T'GO < δT
YES
δT = T'GO
NO
δTINC = δT
2
IPH = I'PH
YES
NO
IPH = 2
NO
YES
CK = 0
CALL ULLDIR TO OBTAIN THE ACTUAL THRUST DIRECTION ŪTA
IPASS = 0
ΔTBURN > 0
YES
4/3
NO
CK = 0
T*BURN = 0
I'PH = -1
NPH = 0
CALL EPERT TO OBTAIN THE PERTURBING ACCELERATION ā PERT DUE TO THE POTENTIAL MODEL AT TIME T
3/3

Figure B-80.- Continued.

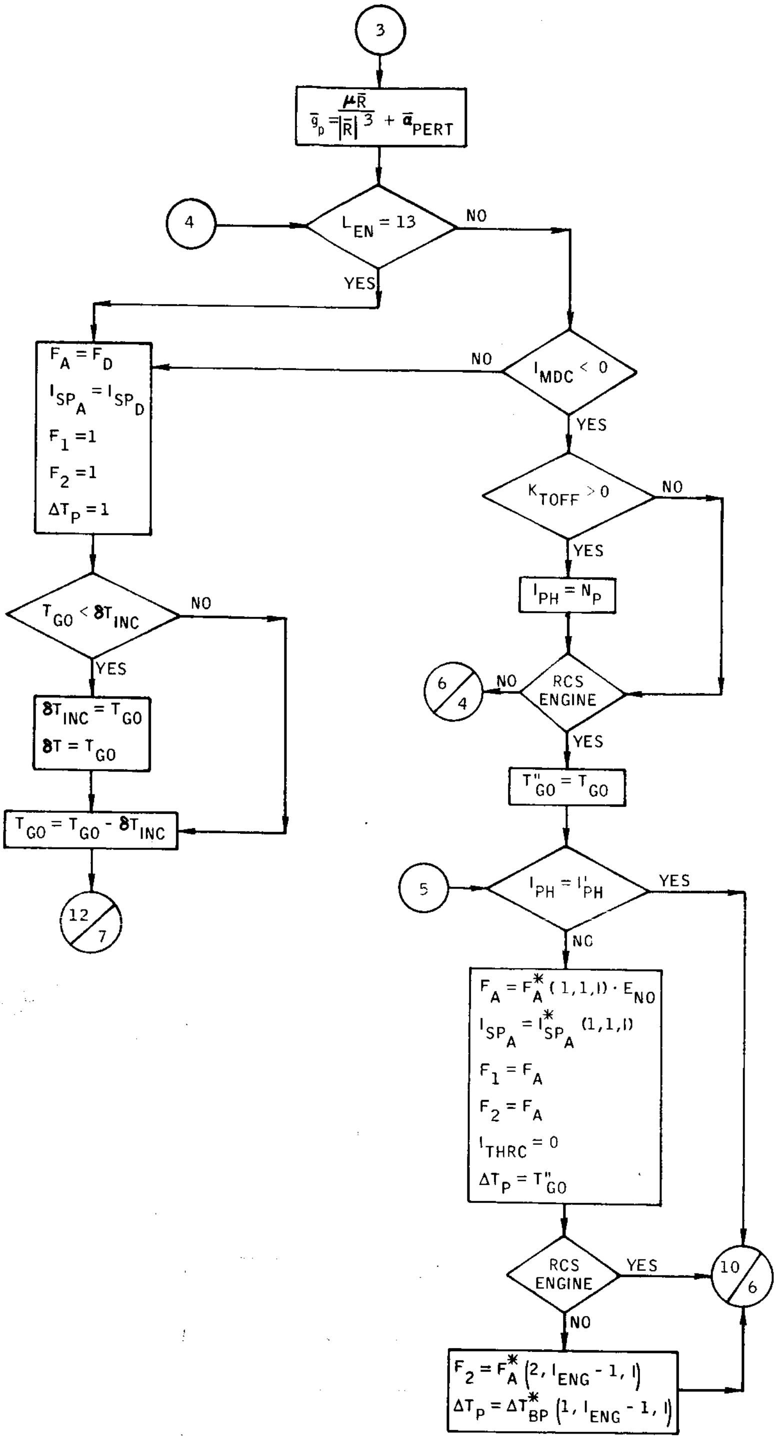


Figure B-80.- Continued.

Page 3 of 9

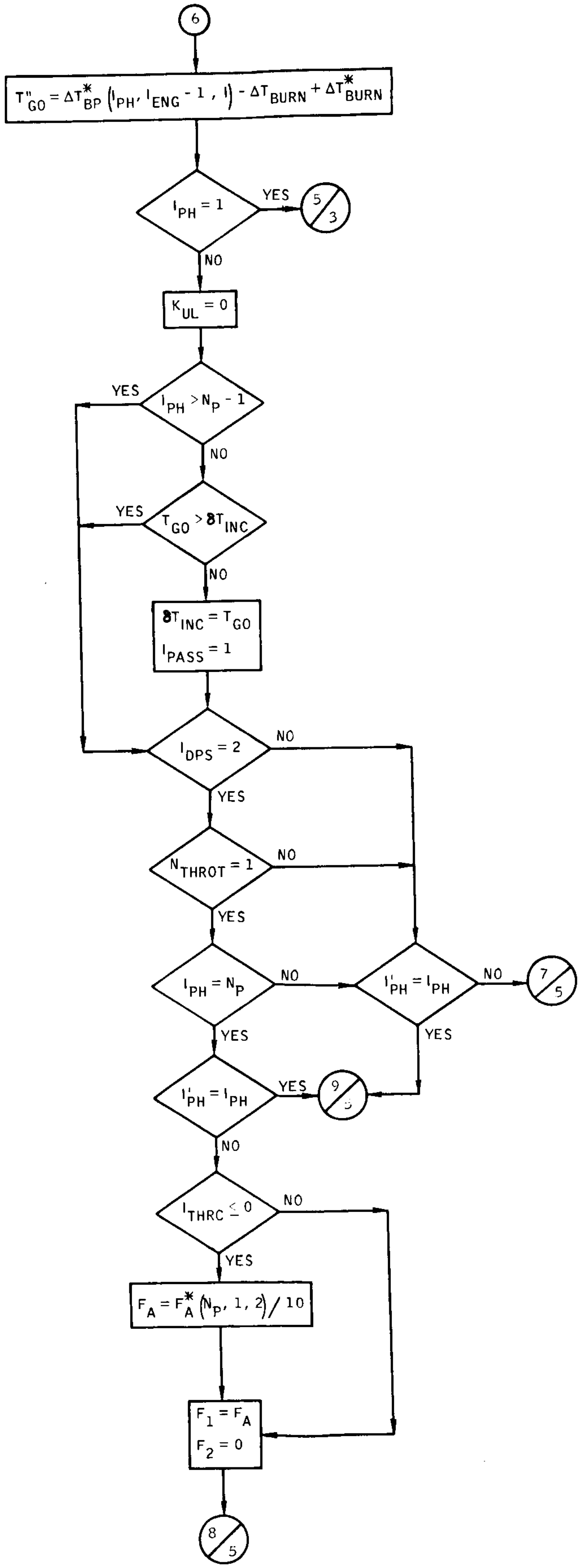
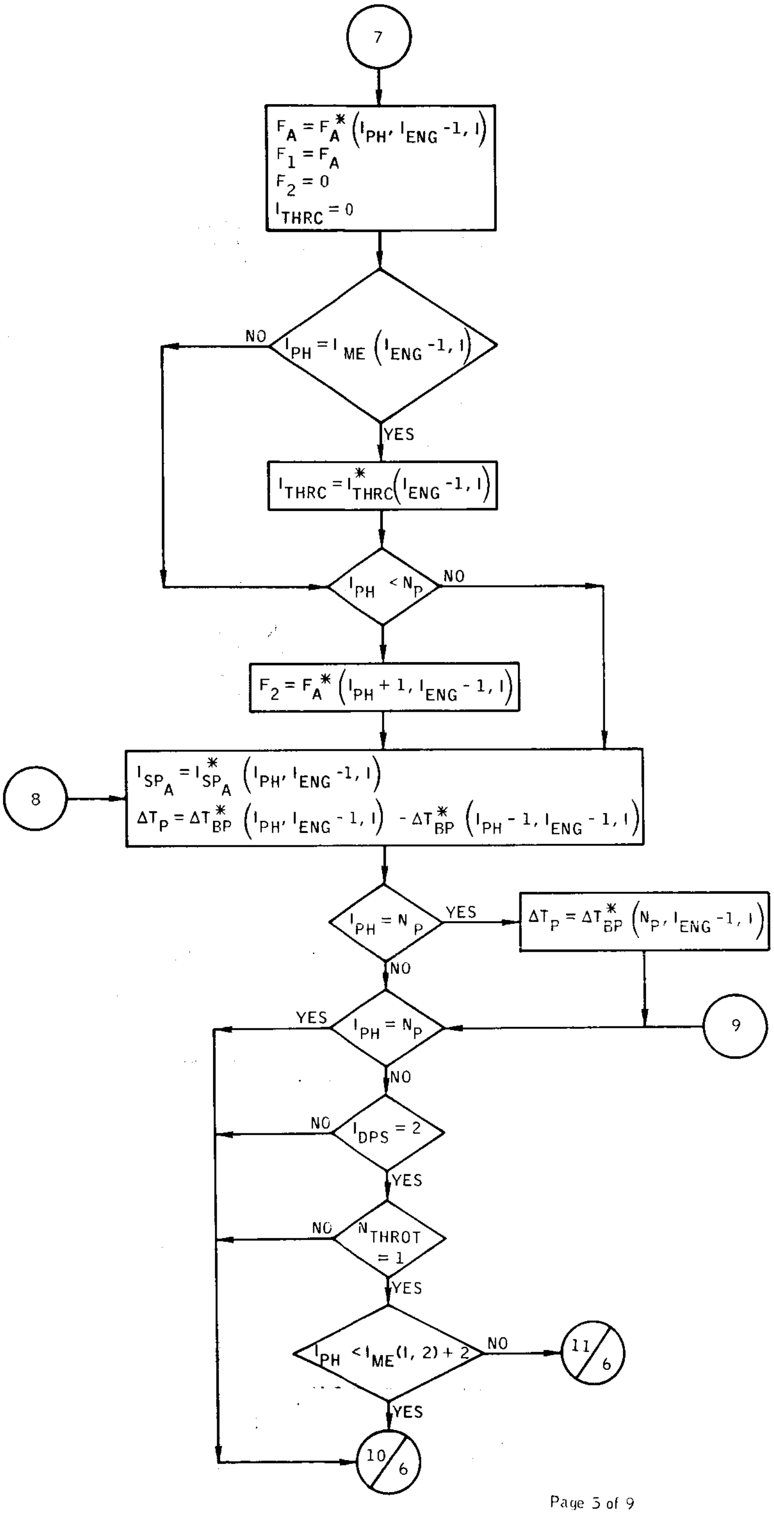


Figure B-80.- Continued.


7
$F_A = F_A^*(I_{PH}, I_{ENG} - 1, I)$
$F_1 = F_A$
$F_2 = 0$
$I_{THRC} = 0$
$I_{PH} = I_{ME}(I_{ENG} - 1, I)$
NO
YES
$I_{THRC} = I_{THRC}^*(I_{ENG} - 1, I)$
$I_{PH} < N_P$
NO
$F_2 = F_A^*(I_{PH} + 1, I_{ENG} - 1, I)$
8
$I_{SP_A} = I_{SP_A}^*(I_{PH}, I_{ENG} - 1, I)$
$\Delta T_P = \Delta T_{BP}^*(I_{PH}, I_{ENG} - 1, I) - \Delta T_{BP}^*(I_{PH} - 1, I_{ENG} - 1, I)$
$I_{PH} = N_P$
YES
$\Delta T_P = \Delta T_{BP}^*(N_P, I_{ENG} - 1, I)$
NO
$I_{PH} = N_P$
YES
9
NO
$I_{DPS} = 2$
NO
YES
$N_{THROT} = 1$
NO
YES
$I_{PH} < I_{ME}(1, 2) + 2$
NO
11/6
YES
10/6

Page 5 of 9

Figure B-80.- Continued.

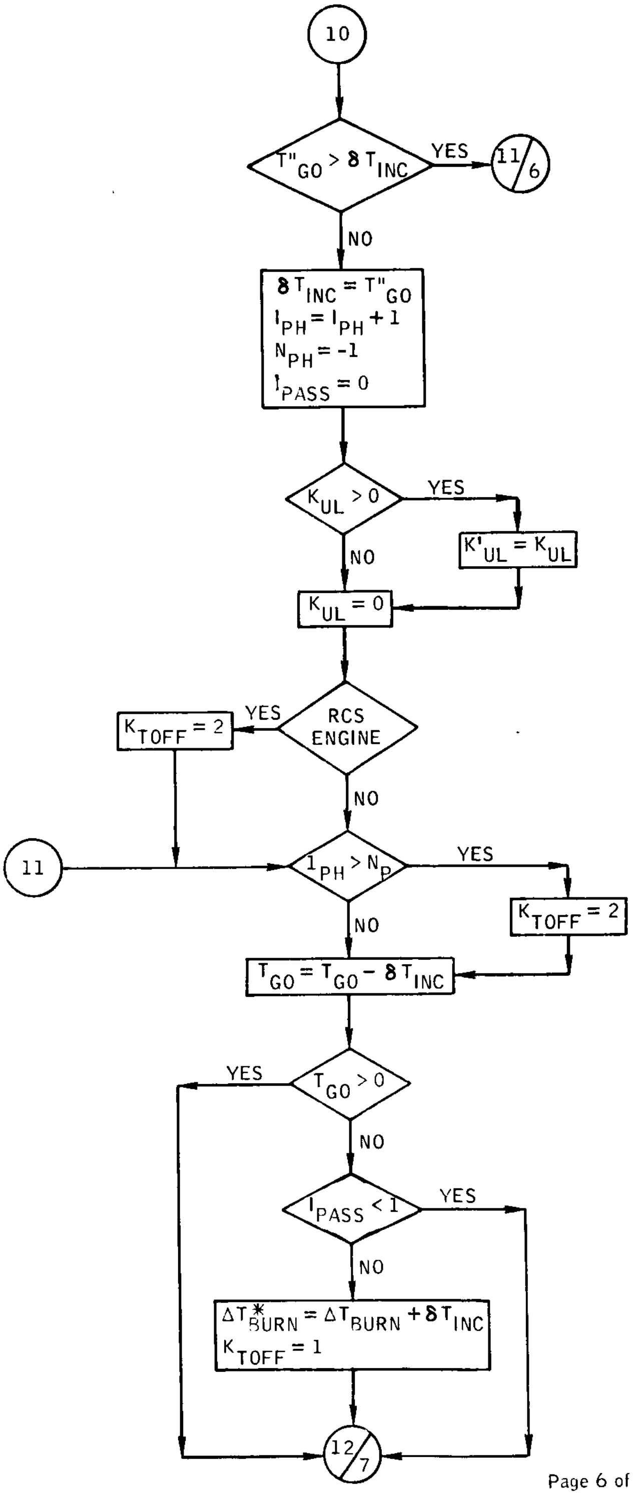
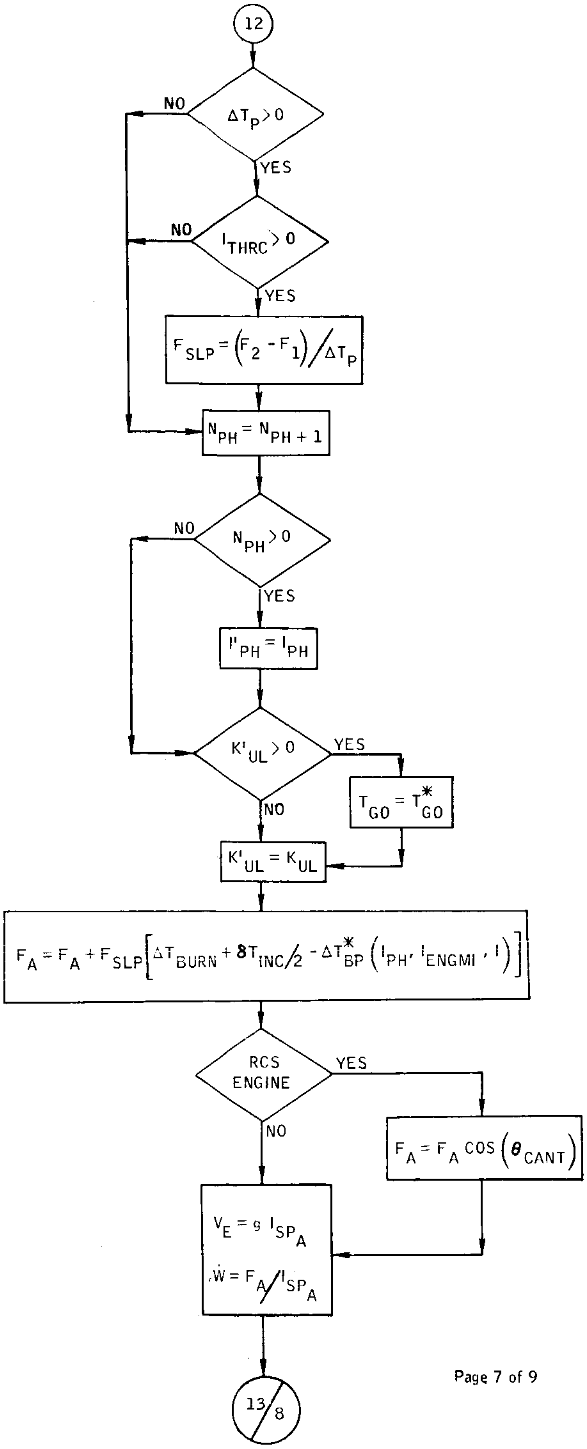
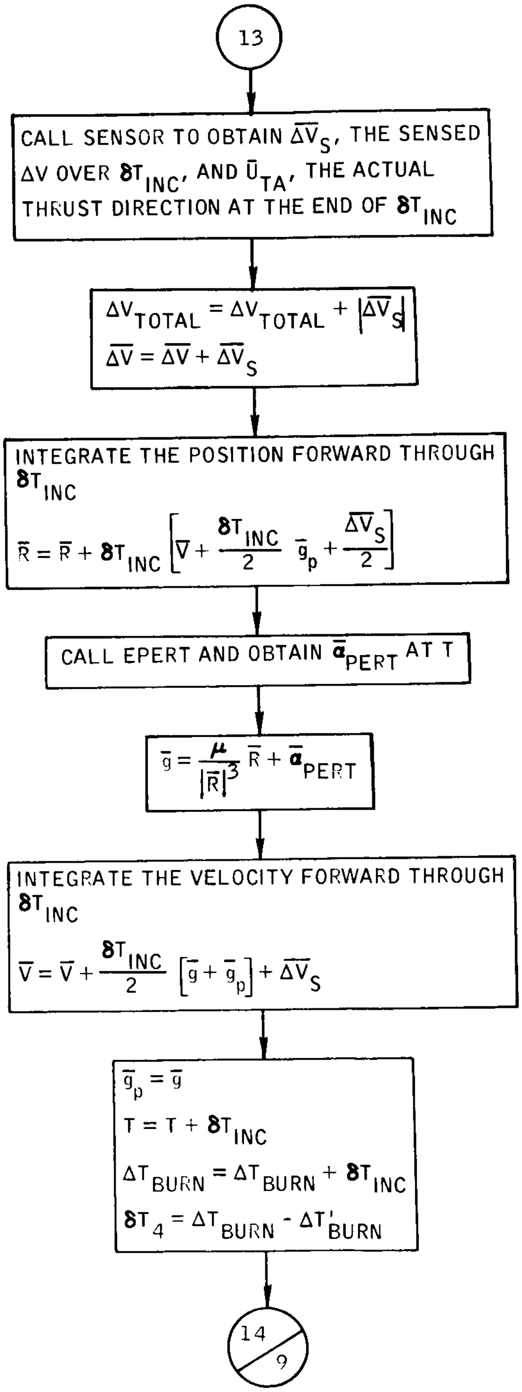


Figure B-80.- Continued.


12
$\Delta T_P > 0$
NO
YES
$I_{THRC} > 0$
NO
YES
$F_{SLP} = (F_2 - F_1) / \Delta T_P$
$N_{PH} = N_{PH} + 1$
$N_{PH} > 0$
NO
YES
$I'_{PH} = I_{PH}$
$K'_{UL} > 0$
YES
NO
$T_{GO} = T^*_{GO}$
$K'_{UL} = K_{UL}$
$F_A = F_A + F_{SLP} \left[\Delta T_{BURN} + \delta T_{INC}/2 - \Delta T^*_{BP} \left(I_{PH}, I_{ENGMI}, 1 \right) \right]$
RCS ENGINE
YES
NO
$F_A = F_A \cos \left(\theta_{CANT} \right)$
$V_E = g\ I_{SP_A}$
$\dot{W} = F_A / I_{SP_A}$
13/8

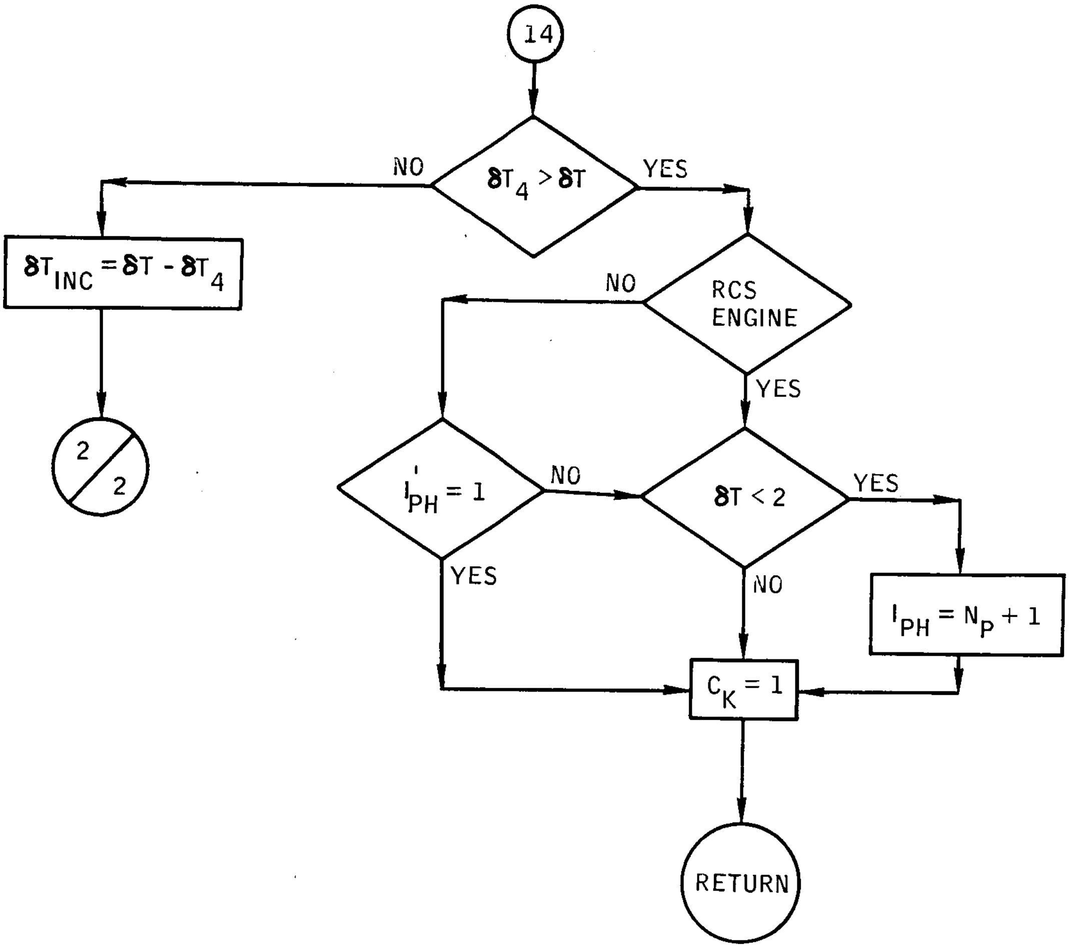
Page 7 of 9

Figure B-80.- Continued.


13
CALL SENSOR TO OBTAIN $\overline{\Delta V}_S$, THE SENSED ΔV OVER δT_{INC}, AND $\bar{U}_{TA}$, THE ACTUAL THRUST DIRECTION AT THE END OF δT_{INC}
$\Delta V_{TOTAL} = \Delta V_{TOTAL} + |\overline{\Delta V}_S|$
$\overline{\Delta V} = \overline{\Delta V} + \overline{\Delta V}_S$
INTEGRATE THE POSITION FORWARD THROUGH δT_{INC}
$\bar{R} = \bar{R} + \delta T_{INC} \left[\bar{V} + \frac{\delta T_{INC}}{2} \bar{g}_p + \frac{\overline{\Delta V}_S}{2} \right]$
CALL EPERT AND OBTAIN $\bar{\alpha}_{PERT}$ AT T
$\bar{g} = \frac{\mu}{|\bar{R}|^3} \bar{R} + \bar{\alpha}_{PERT}$
INTEGRATE THE VELOCITY FORWARD THROUGH δT_{INC}
$\bar{V} = \bar{V} + \frac{\delta T_{INC}}{2} \left[\bar{g} + \bar{g}_p \right] + \overline{\Delta V}_S$
$\bar{g}_p = \bar{g}$
$T = T + \delta T_{INC}$
$\Delta T_{BURN} = \Delta T_{BURN} + \delta T_{INC}$
$\delta T_4 = \Delta T_{BURN} - \Delta T'_{BURN}$
14
9

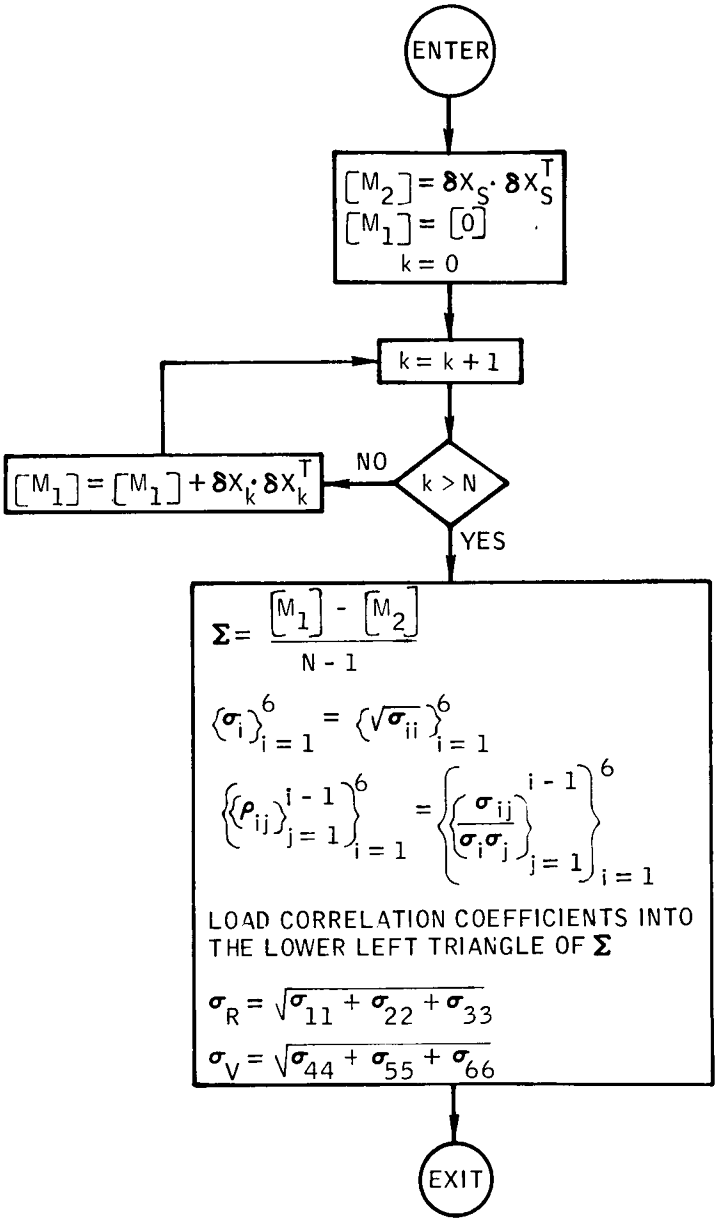
Page 8 of 9

Figure B-80.- Continued.



Page 9 of 9

Figure B-80.- Concluded.


ENTER
$[M_2] = \delta X_S \cdot \delta X_S^T$
$[M_1] = [0]$
$k = 0$
$k = k + 1$
$k > N$
NO
$[M_1] = [M_1] + \delta X_k \cdot \delta X_k^T$
YES
$\Sigma = \frac{[M_1] - [M_2]}{N - 1}$
$\{\sigma_i\}_{i=1}^{6} = \{\sqrt{\sigma_{ii}}\}_{i=1}^{6}$
$\left\{\{\rho_{ij}\}_{j=1}^{i-1}\right\}_{i=1}^{6} = \left\{\left\{\frac{\sigma_{ij}}{\sigma_i \sigma_j}\right\}_{j=1}^{i-1}\right\}_{i=1}^{6}$
LOAD CORRELATION COEFFICIENTS INTO THE LOWER LEFT TRIANGLE OF Σ
$\sigma_R = \sqrt{\sigma_{11} + \sigma_{22} + \sigma_{33}}$
$\sigma_V = \sqrt{\sigma_{44} + \sigma_{55} + \sigma_{66}}$
EXIT

Page 1 of 1

Figure B-81.- Flow chart of subroutine SAMCOV.

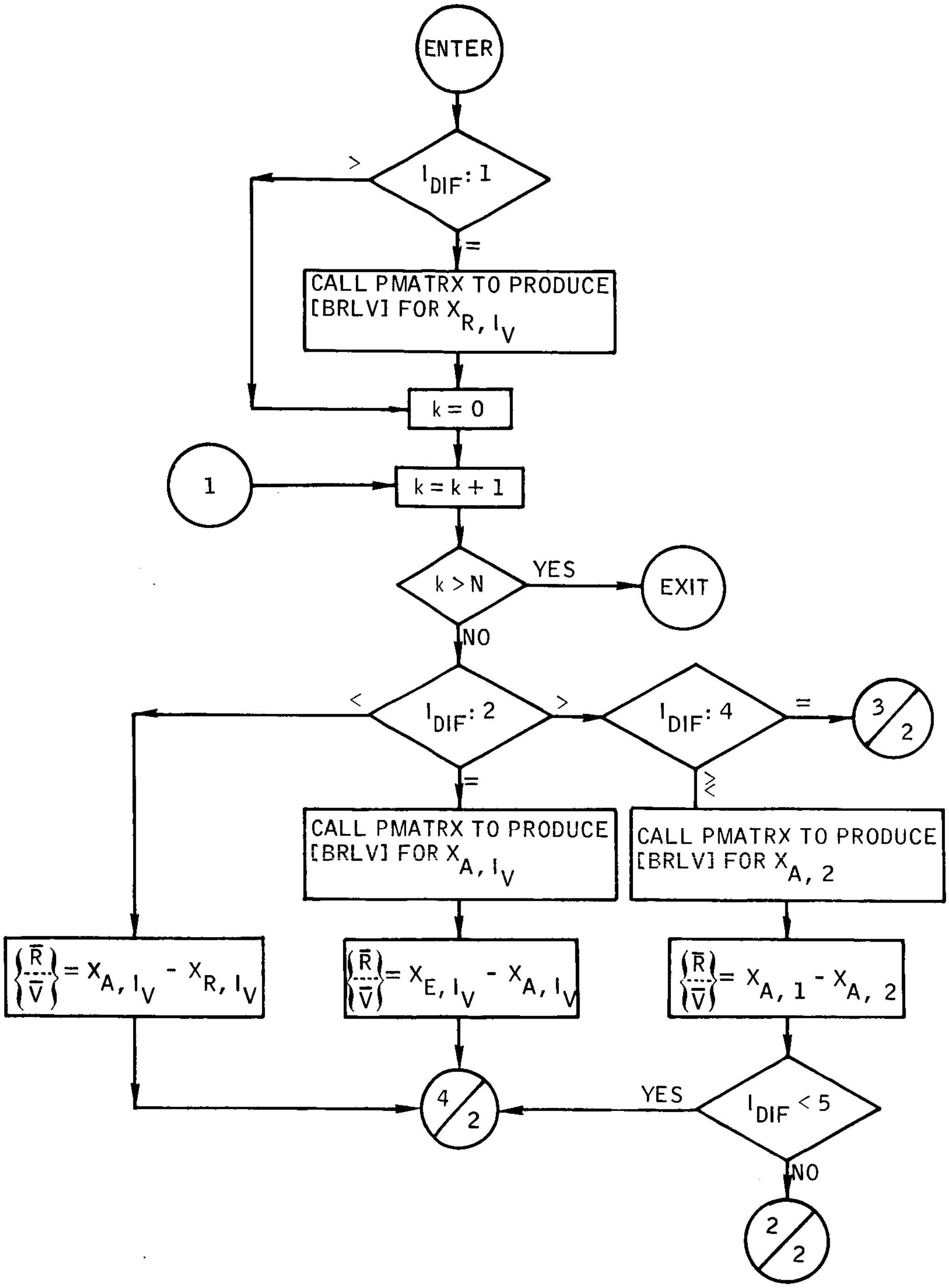

ENTER
$I_{DIF}: 1$
>
=
CALL PMATRX TO PRODUCE [BRLV] FOR X_{R, I_V}
k = 0
1
k = k + 1
k > N
YES
EXIT
NO
$I_{DIF}: 2$
<
>
=
$I_{DIF}: 4$
=
3/2
≷
CALL PMATRX TO PRODUCE [BRLV] FOR X_{A, I_V}
CALL PMATRX TO PRODUCE [BRLV] FOR $X_{A, 2}$
$\left\{\frac{\bar{R}}{\bar{V}}\right\} = X_{A, I_V} - X_{R, I_V}$
$\left\{\frac{\bar{R}}{\bar{V}}\right\} = X_{E, I_V} - X_{A, I_V}$
$\left\{\frac{\bar{R}}{\bar{V}}\right\} = X_{A, 1} - X_{A, 2}$
4/2
YES
$I_{DIF} < 5$
NO
2/2

Figure B-82.- Flow chart of subroutine SAMDIF. Page 1 of 2

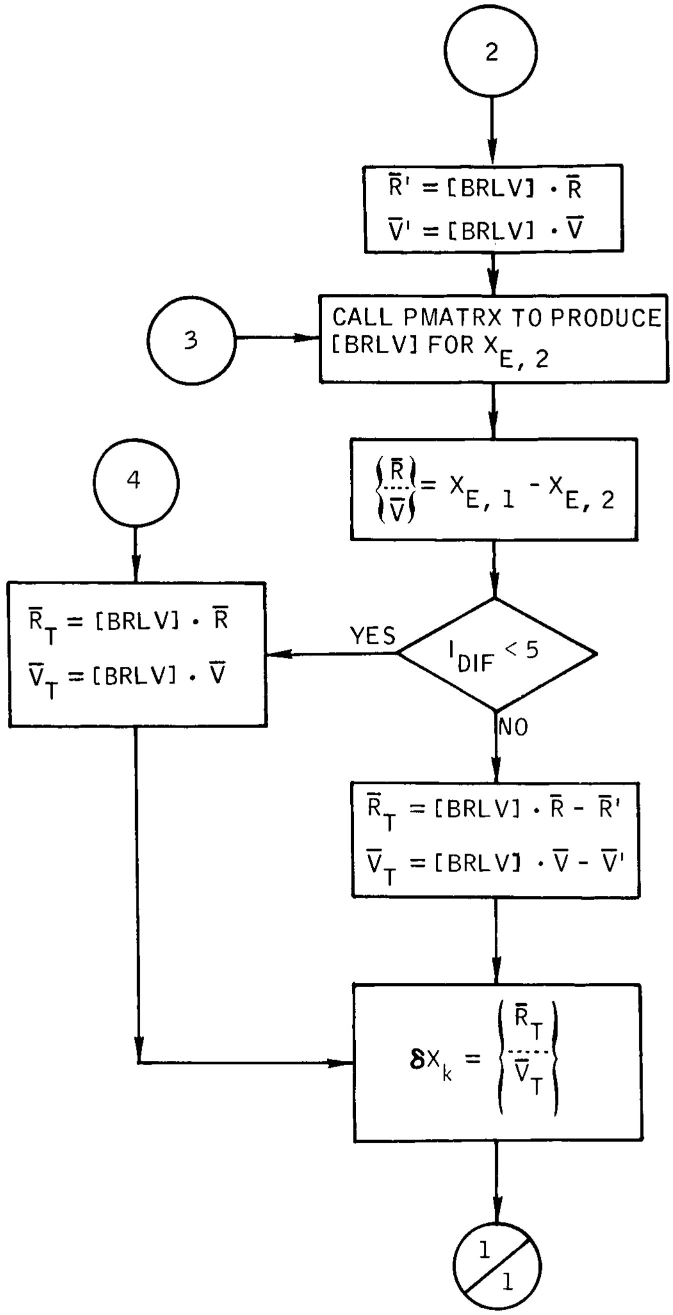
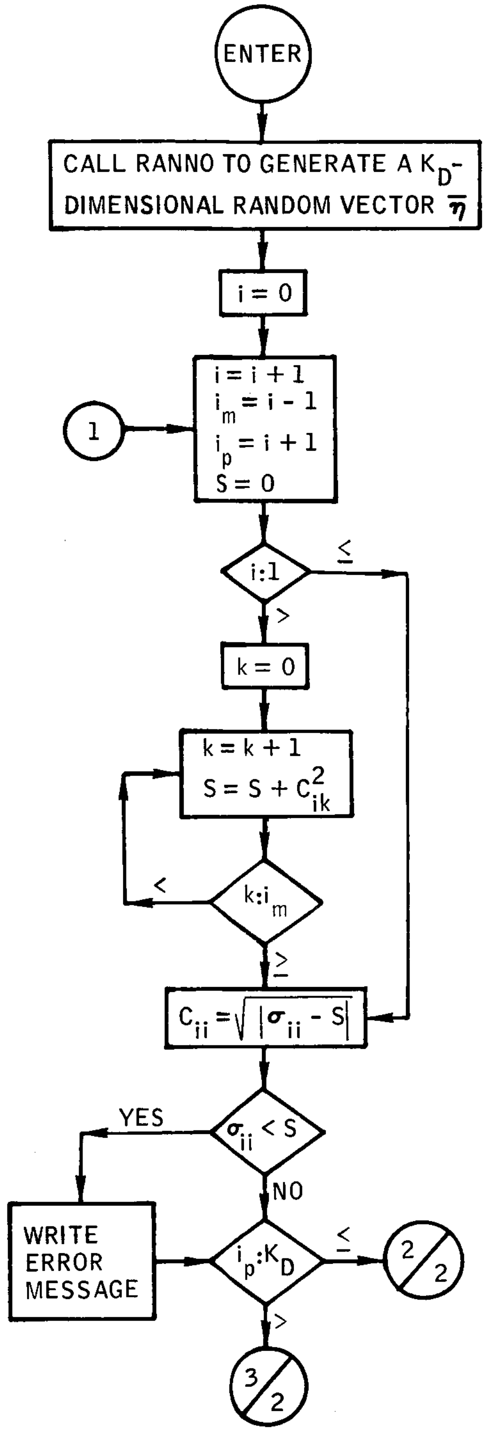

2
$\bar{R}' = [BRLV] \cdot \bar{R}$
$\bar{V}' = [BRLV] \cdot \bar{V}$
3
CALL PMATRX TO PRODUCE [BRLV] FOR $X_{E,2}$
4
$\left\{\frac{\bar{R}}{\bar{V}}\right\} = X_{E,1} - X_{E,2}$
$I_{DIF} < 5$
YES
NO
$\bar{R}_T = [BRLV] \cdot \bar{R}$
$\bar{V}_T = [BRLV] \cdot \bar{V}$
$\bar{R}_T = [BRLV] \cdot \bar{R} - \bar{R}'$
$\bar{V}_T = [BRLV] \cdot \bar{V} - \bar{V}'$
$\delta X_k = \left\{\frac{\bar{R}_T}{\bar{V}_T}\right\}$
1
1

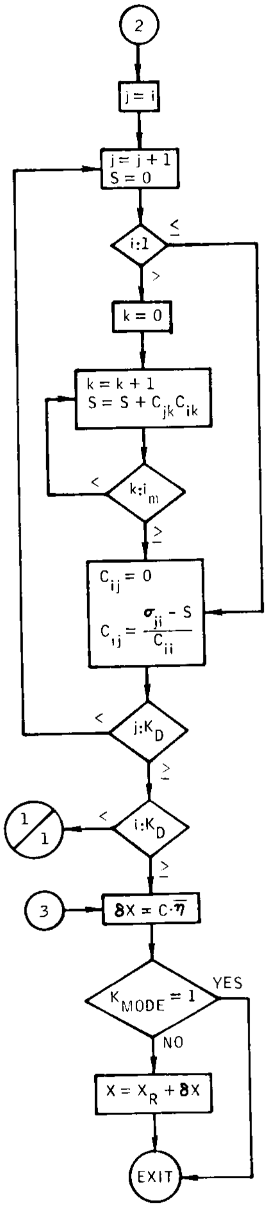
Figure B-82.- Concluded.

Page 2 of 2



Page 1 of 2

Figure B-83.- Flow chart of subroutine SAMPLE.



Page 2 of 2

Figure B-83.- Concluded.

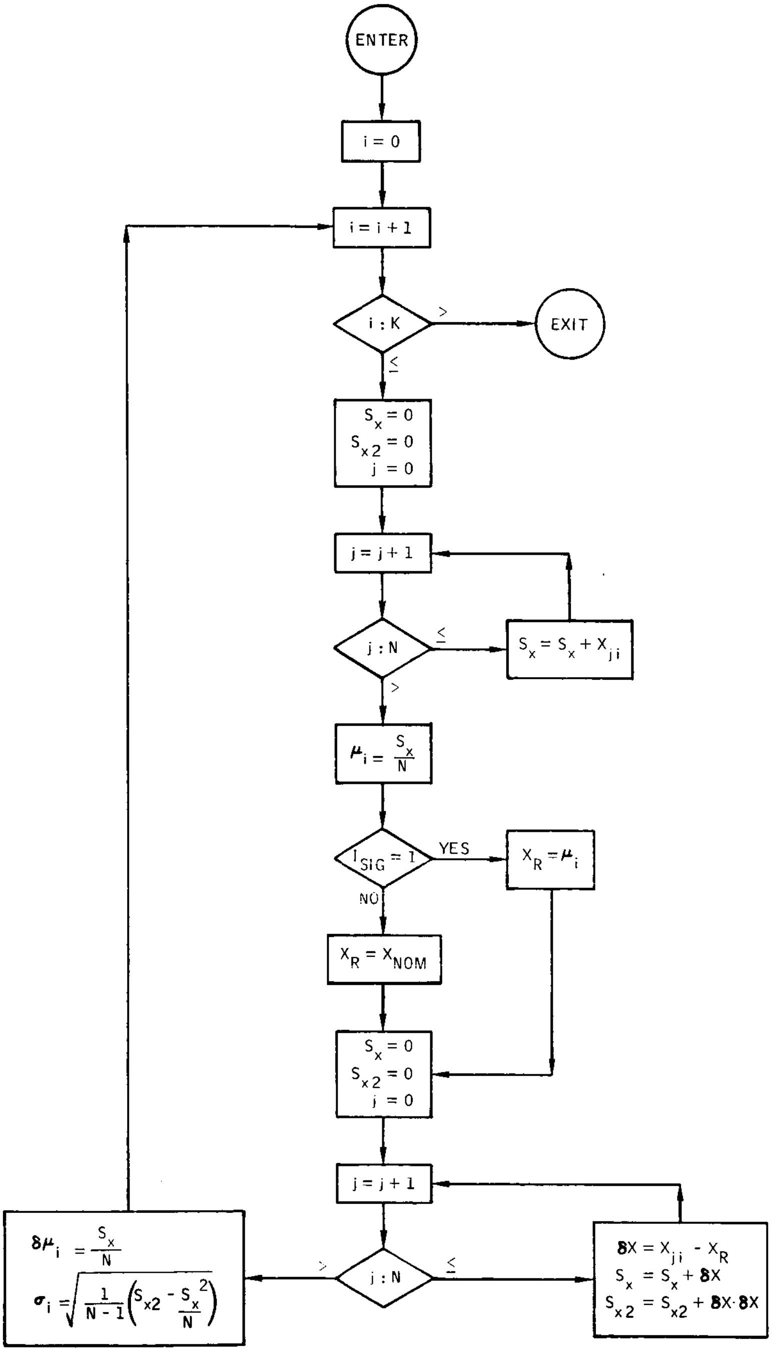
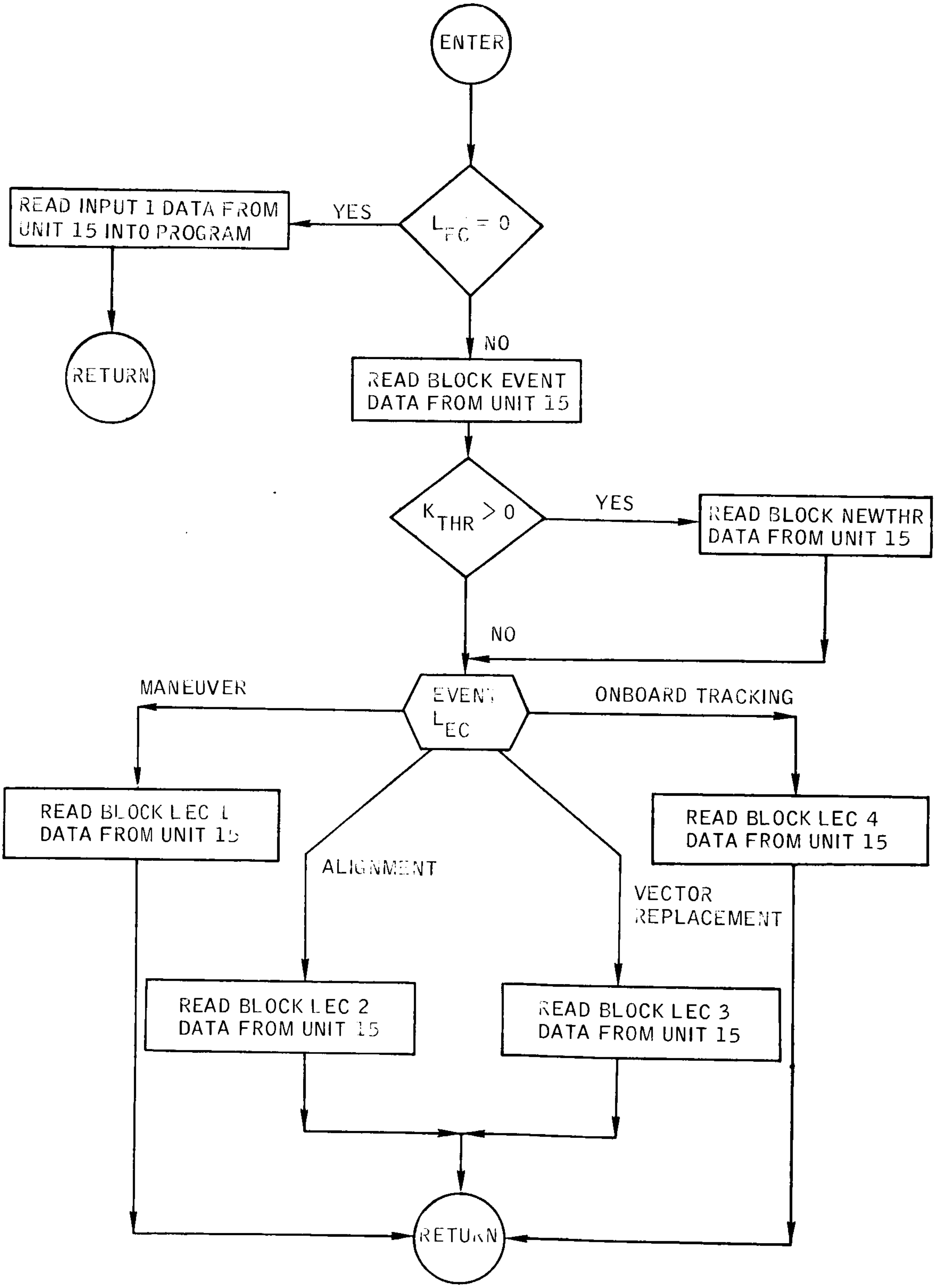


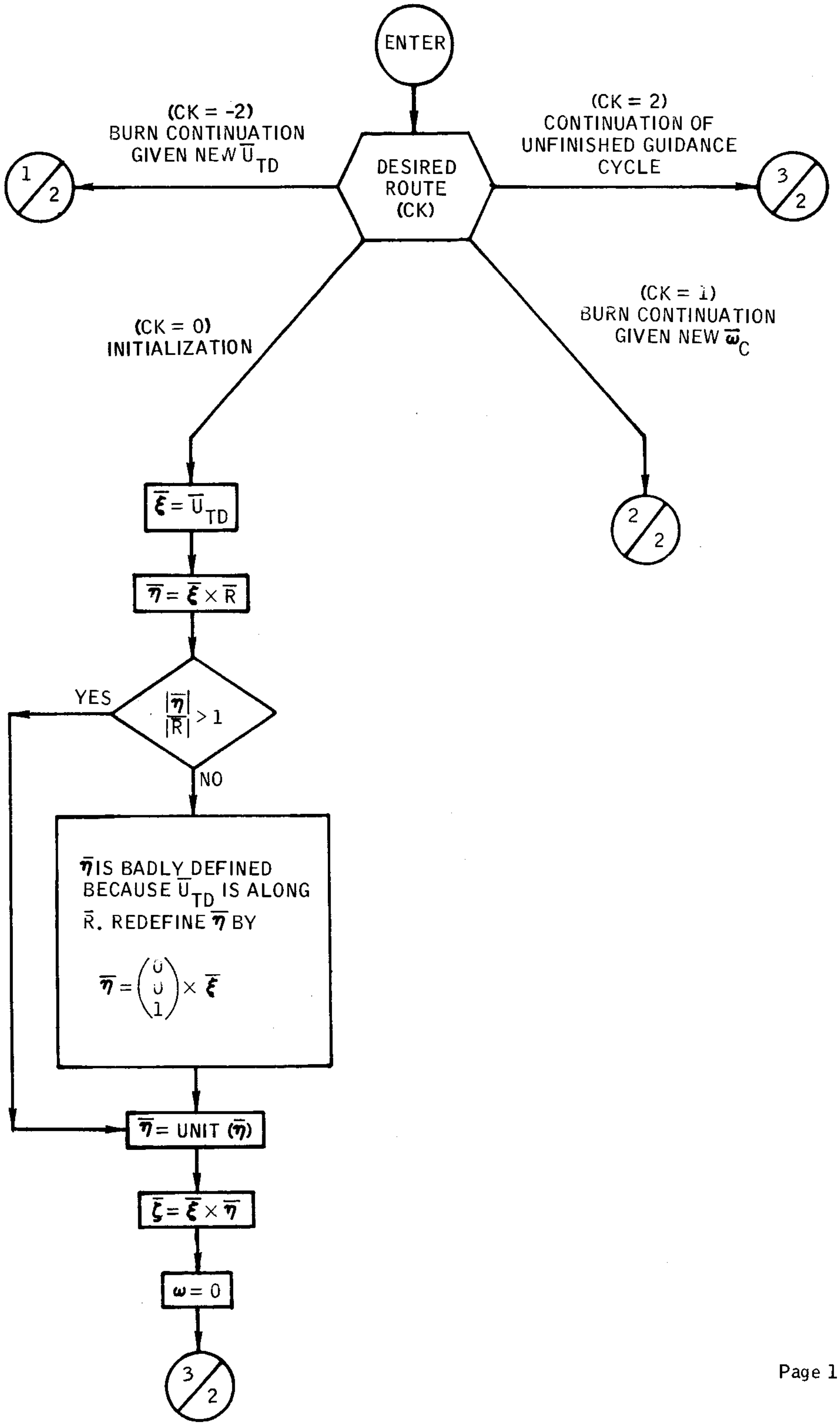
Figure B-34.- Flow chart of subroutine SAMSIG.

Page 1 of 1


ENTER
$L_{FC} = 0$
YES
READ INPUT 1 DATA FROM UNIT 15 INTO PROGRAM
RETURN
NO
READ BLOCK EVENT DATA FROM UNIT 15
$K_{THR} > 0$
YES
READ BLOCK NEWTHR DATA FROM UNIT 15
NO
EVENT L_{EC}
MANEUVER
ONBOARD TRACKING
READ BLOCK LEC 1 DATA FROM UNIT 15
ALIGNMENT
VECTOR REPLACEMENT
READ BLOCK LEC 4 DATA FROM UNIT 15
READ BLOCK LEC 2 DATA FROM UNIT 15
READ BLOCK LEC 3 DATA FROM UNIT 15
RETURN

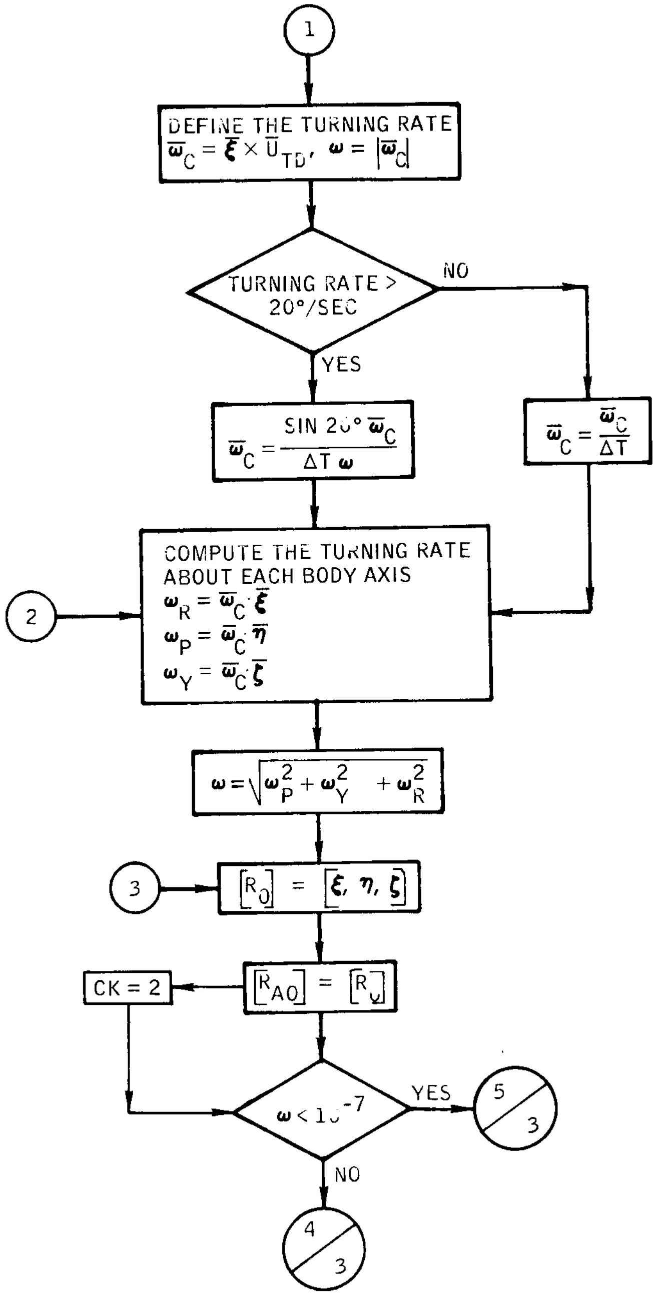
Page 1 of 1

Figure B-85.- Flow chart of subroutine SAVIP 2.


ENTER
DESIRED ROUTE (CK)
(CK = -2) BURN CONTINUATION GIVEN NEW $\bar{U}_{TD}$
1/2
(CK = 2) CONTINUATION OF UNFINISHED GUIDANCE CYCLE
3/2
(CK = 0) INITIALIZATION
(CK = 1) BURN CONTINUATION GIVEN NEW $\bar{\omega}_C$
2/2
$\bar{\xi} = \bar{U}_{TD}$
$\bar{\eta} = \bar{\xi} \times \bar{R}$
$\frac{|\bar{\eta}|}{|\bar{R}|} > 1$
YES
NO
$\bar{\eta}$ IS BADLY DEFINED BECAUSE $\bar{U}_{TD}$ IS ALONG $\bar{R}$. REDEFINE $\bar{\eta}$ BY
$\bar{\eta} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \times \bar{\xi}$
$\bar{\eta}$ = UNIT ($\bar{\eta}$)
$\bar{\zeta} = \bar{\xi} \times \bar{\eta}$
$\omega = 0$
3/2

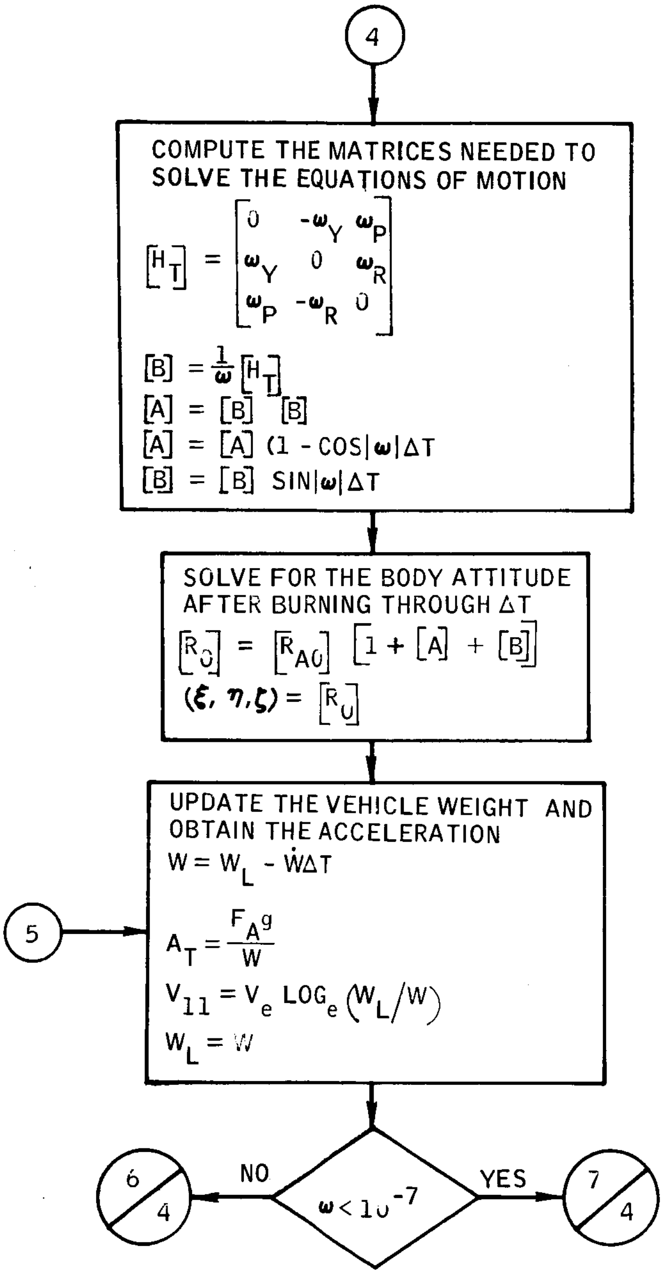
Page 1 of 4

Figure B-86.- Flow chart of subroutine SENSOR.


1
DEFINE THE TURNING RATE
$\overline{\omega}_C = \overline{\xi} \times \overline{U}_{TD}$, $\omega = |\overline{\omega}_C|$
TURNING RATE > 20°/SEC
NO
YES
$\overline{\omega}_C = \frac{\text{SIN } 20° \ \overline{\omega}_C}{\Delta T \ \omega}$
$\overline{\omega}_C = \frac{\overline{\omega}_C}{\Delta T}$
2
COMPUTE THE TURNING RATE ABOUT EACH BODY AXIS
$\omega_R = \overline{\omega}_C \cdot \overline{\xi}$
$\omega_P = \overline{\omega}_C \cdot \overline{\eta}$
$\omega_Y = \overline{\omega}_C \cdot \overline{\zeta}$
$\omega = \sqrt{\omega_P^2 + \omega_Y^2 + \omega_R^2}$
3
$[R_0] = [\xi, \eta, \zeta]$
CK = 2
$[R_{A0}] = [R_J]$
$\omega < 10^{-7}$
YES
NO
5/3
4/3

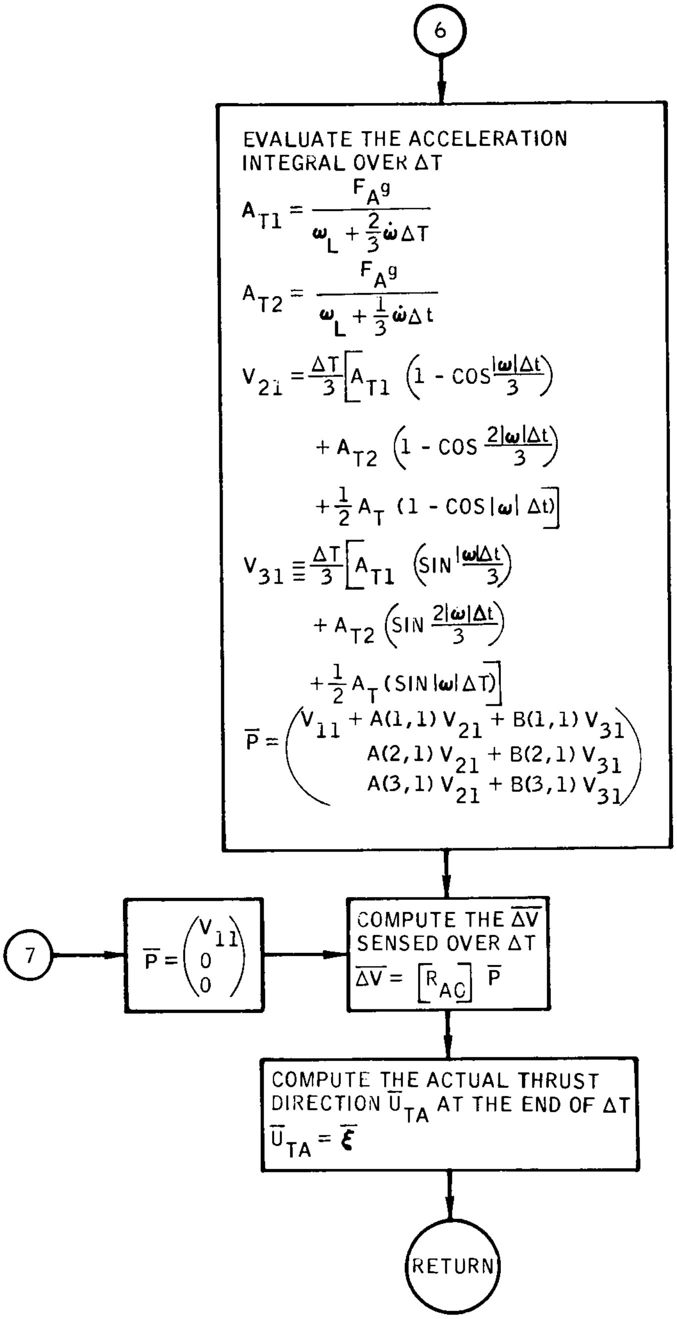
Page 2 of 4

Figure B-85.- Continued.


4
COMPUTE THE MATRICES NEEDED TO SOLVE THE EQUATIONS OF MOTION
$[H_T] = \begin{bmatrix} 0 & -\omega_Y & \omega_P \\ \omega_Y & 0 & \omega_R \\ \omega_P & -\omega_R & 0 \end{bmatrix}$
$[B] = \frac{1}{\omega}[H_T]$
$[A] = [B] \quad [B]$
$[A] = [A] \ (1 - \text{COS}|\omega|\Delta T$
$[B] = [B] \ \text{SIN}|\omega|\Delta T$
SOLVE FOR THE BODY ATTITUDE AFTER BURNING THROUGH ΔT
$[R_0] = [R_{A0}] \ [1 + [A] + [B]]$
$(\xi, \eta, \zeta) = [R_0]$
UPDATE THE VEHICLE WEIGHT AND OBTAIN THE ACCELERATION
$W = W_L - \dot{W}\Delta T$
5
$A_T = \frac{F_A g}{W}$
$V_{11} = V_e \ \text{LOG}_e (W_L/W)$
$W_L = W$
$\omega < 10^{-7}$
NO
YES
6
4
7
4

Page 3 of 4

Figure B-86.- Continued.



$$A_{T1} = \frac{F_A g}{\omega_L + \frac{2}{3}\dot{\omega}\Delta T}$$

$$A_{T2} = \frac{F_A g}{\omega_L + \frac{1}{3}\dot{\omega}\Delta t}$$

$$V_{21} = \frac{\Delta T}{3}\left[A_{T1}\left(1 - \cos\frac{|\omega|\Delta t}{3}\right) + A_{T2}\left(1 - \cos\frac{2|\omega|\Delta t}{3}\right) + \frac{1}{2}A_T\left(1 - \cos|\omega|\Delta t\right)\right]$$

$$V_{31} \equiv \frac{\Delta T}{3}\left[A_{T1}\left(\text{SIN}\frac{|\omega|\Delta t}{3}\right) + A_{T2}\left(\text{SIN}\frac{2|\dot{\omega}|\Delta t}{3}\right) + \frac{1}{2}A_T\left(\text{SIN}|\omega|\Delta T\right)\right]$$

$$\overline{P} = \begin{pmatrix} V_{11} + A(1,1)V_{21} + B(1,1)V_{31} \\ A(2,1)V_{21} + B(2,1)V_{31} \\ A(3,1)V_{21} + B(3,1)V_{31} \end{pmatrix}$$

$$\overline{P} = \begin{pmatrix} V_{11} \\ 0 \\ 0 \end{pmatrix}$$

$$\overline{\Delta V} = \left[R_{AC}\right]\overline{P}$$

COMPUTE THE ACTUAL THRUST DIRECTION $\overline{U}_{TA}$ AT THE END OF ΔT

$$\overline{U}_{TA} = \overline{\xi}$$

Page 4 of 4

Figure B-86.- Concluded.

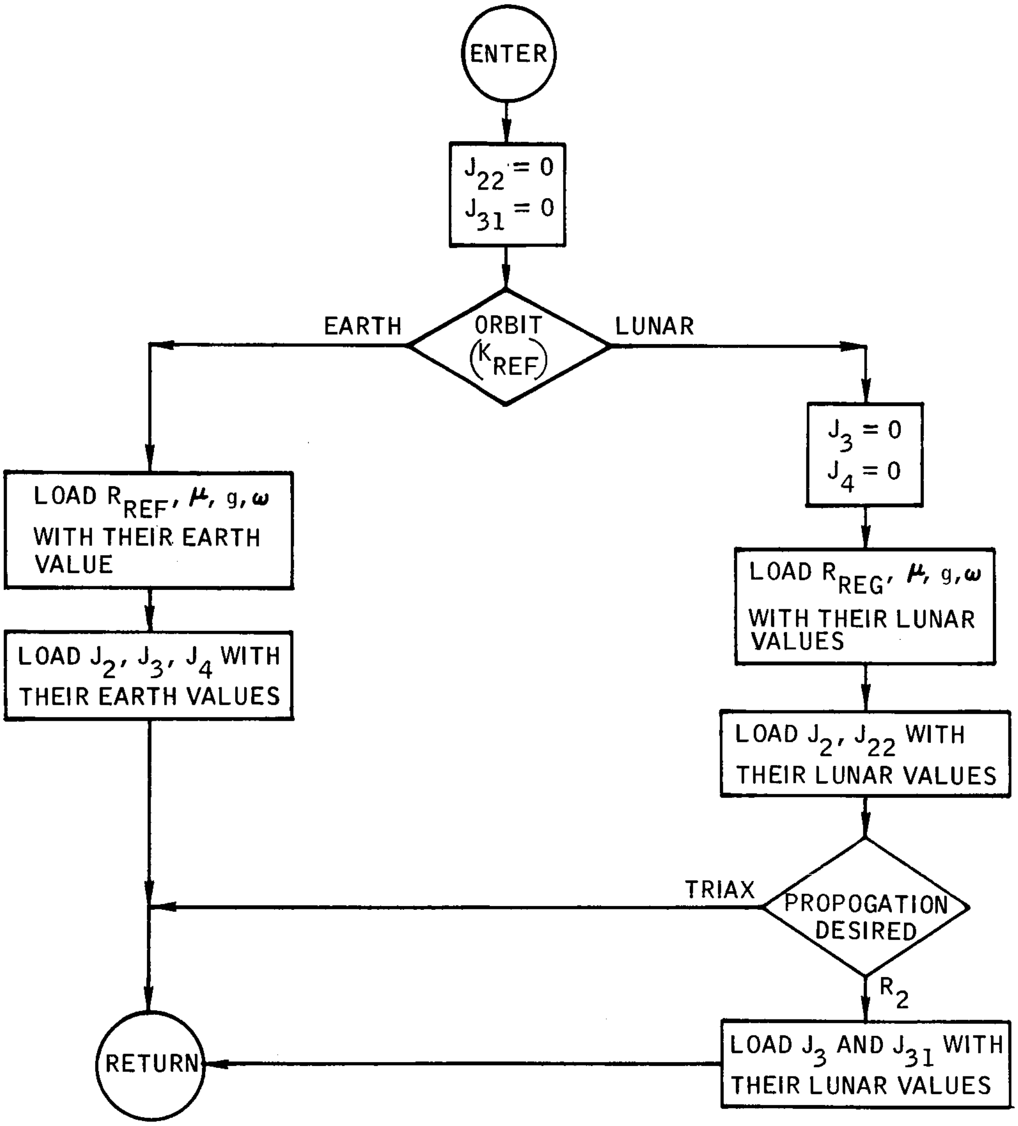
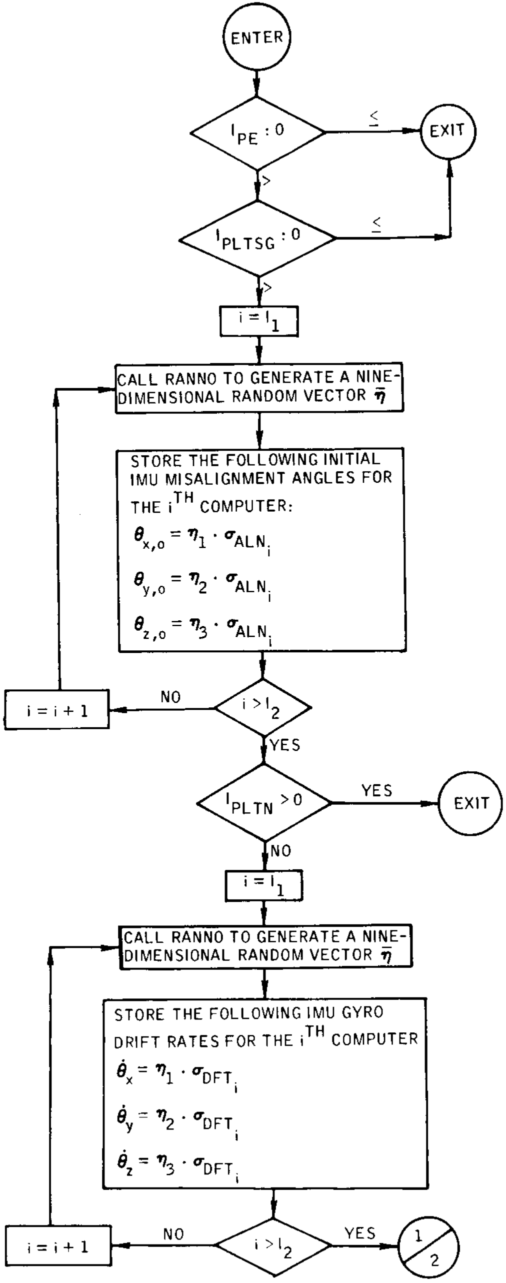

ENTER
$J_{22} = 0$
$J_{31} = 0$
EARTH
ORBIT (K_{REF})
LUNAR
$J_3 = 0$
$J_4 = 0$
LOAD R_{REF}, μ, g, ω WITH THEIR EARTH VALUE
LOAD R_{REG}, μ, g, ω WITH THEIR LUNAR VALUES
LOAD J_2, J_3, J_4 WITH THEIR EARTH VALUES
LOAD J_2, J_{22} WITH THEIR LUNAR VALUES
TRIAX
PROPOGATION DESIRED
R_2
RETURN
LOAD J_3 AND J_{31} WITH THEIR LUNAR VALUES

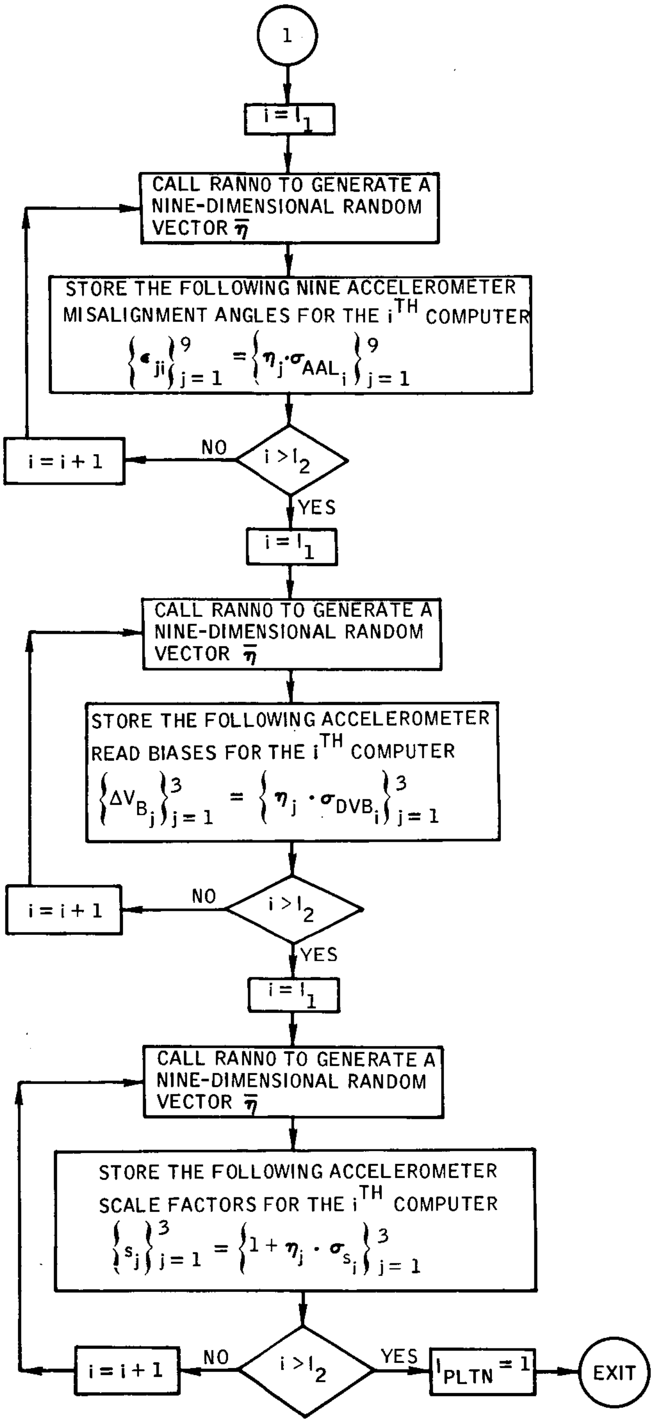
Figure B-87.- Flow chart of subroutine SETPC.

Page 1 of 1


ENTER
$I_{PE} : 0$
≤
EXIT
>
$I_{PLTSG} : 0$
≤
>
$i = I_1$
CALL RANNO TO GENERATE A NINE-DIMENSIONAL RANDOM VECTOR $\bar{\eta}$
STORE THE FOLLOWING INITIAL IMU MISALIGNMENT ANGLES FOR THE i^{TH} COMPUTER:
$\theta_{x,o} = \eta_1 \cdot \sigma_{ALN_i}$
$\theta_{y,o} = \eta_2 \cdot \sigma_{ALN_i}$
$\theta_{z,o} = \eta_3 \cdot \sigma_{ALN_i}$
$i = i + 1$
NO
$i > I_2$
YES
$I_{PLTN} > 0$
YES
EXIT
NO
$i = I_1$
CALL RANNO TO GENERATE A NINE-DIMENSIONAL RANDOM VECTOR $\bar{\eta}$
STORE THE FOLLOWING IMU GYRO DRIFT RATES FOR THE i^{TH} COMPUTER
$\dot{\theta}_x = \eta_1 \cdot \sigma_{DFT_i}$
$\dot{\theta}_y = \eta_2 \cdot \sigma_{DFT_i}$
$\dot{\theta}_z = \eta_3 \cdot \sigma_{DFT_i}$
$i = i + 1$
NO
$i > I_2$
YES
1
2

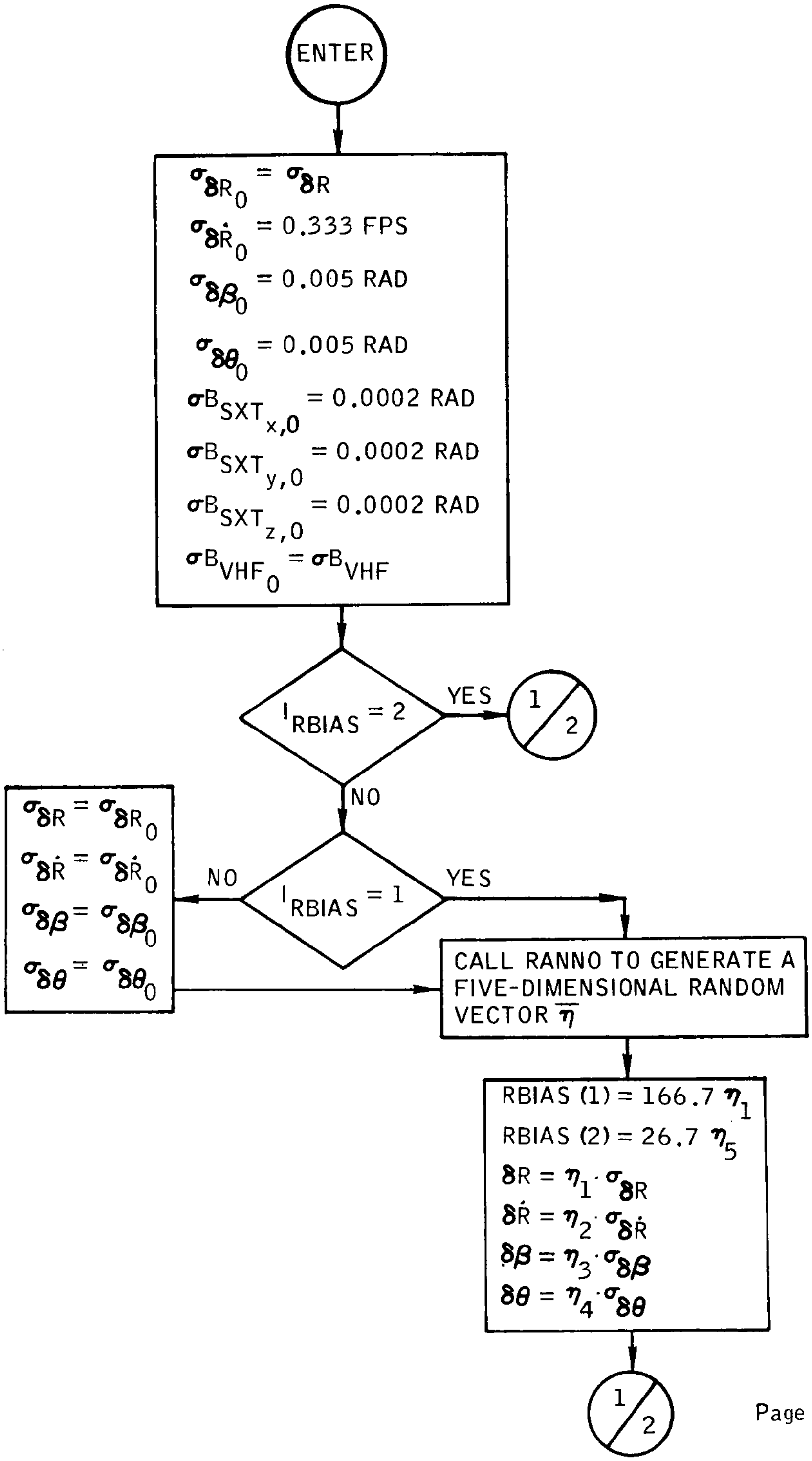
Page 1 of 2

Figure B-38.- Flow chart of subroutine SETPLT.


1
$i = I_1$
CALL RANNO TO GENERATE A NINE-DIMENSIONAL RANDOM VECTOR $\bar{\eta}$
STORE THE FOLLOWING NINE ACCELEROMETER MISALIGNMENT ANGLES FOR THE i^{TH} COMPUTER
$\left\{\epsilon_{ji}\right\}_{j=1}^{9} = \left\{\eta_j \cdot \sigma_{AAL_i}\right\}_{j=1}^{9}$
$i > I_2$
NO
$i = i + 1$
YES
$i = I_1$
CALL RANNO TO GENERATE A NINE-DIMENSIONAL RANDOM VECTOR $\bar{\eta}$
STORE THE FOLLOWING ACCELEROMETER READ BIASES FOR THE i^{TH} COMPUTER
$\left\{\Delta V_{B_j}\right\}_{j=1}^{3} = \left\{\eta_j \cdot \sigma_{DVB_i}\right\}_{j=1}^{3}$
$i > I_2$
NO
$i = i + 1$
YES
$i = I_1$
CALL RANNO TO GENERATE A NINE-DIMENSIONAL RANDOM VECTOR $\bar{\eta}$
STORE THE FOLLOWING ACCELEROMETER SCALE FACTORS FOR THE i^{TH} COMPUTER
$\left\{s_j\right\}_{j=1}^{3} = \left\{1 + \eta_j \cdot \sigma_{s_i}\right\}_{j=1}^{3}$
$i = i + 1$
NO
$i > I_2$
YES
$I_{PLTN} = 1$
EXIT

Page 2 of 2

Figure B-88.- Concluded.


ENTER
$\sigma_{\delta R_0} = \sigma_{\delta R}$
$\sigma_{\delta \dot{R}_0} = 0.333$ FPS
$\sigma_{\delta \beta_0} = 0.005$ RAD
$\sigma_{\delta \theta_0} = 0.005$ RAD
$\sigma B_{SXT_{x,0}} = 0.0002$ RAD
$\sigma B_{SXT_{y,0}} = 0.0002$ RAD
$\sigma B_{SXT_{z,0}} = 0.0002$ RAD
$\sigma B_{VHF_0} = \sigma B_{VHF}$
$I_{RBIAS} = 2$
YES
1/2
NO
$I_{RBIAS} = 1$
NO
YES
$\sigma_{\delta R} = \sigma_{\delta R_0}$
$\sigma_{\delta \dot{R}} = \sigma_{\delta \dot{R}_0}$
$\sigma_{\delta \beta} = \sigma_{\delta \beta_0}$
$\sigma_{\delta \theta} = \sigma_{\delta \theta_0}$
CALL RANNO TO GENERATE A FIVE-DIMENSIONAL RANDOM VECTOR $\overline{\eta}$
RBIAS (1) = 166.7 η_1
RBIAS (2) = 26.7 η_5
$\delta R = \eta_1 \cdot \sigma_{\delta R}$
$\delta \dot{R} = \eta_2 \cdot \sigma_{\delta \dot{R}}$
$\delta \beta = \eta_3 \cdot \sigma_{\delta \beta}$
$\delta \theta = \eta_4 \cdot \sigma_{\delta \theta}$
1/2

Page 1 of 2

Figure B-89.- Flow chart of subroutine SETRBS.

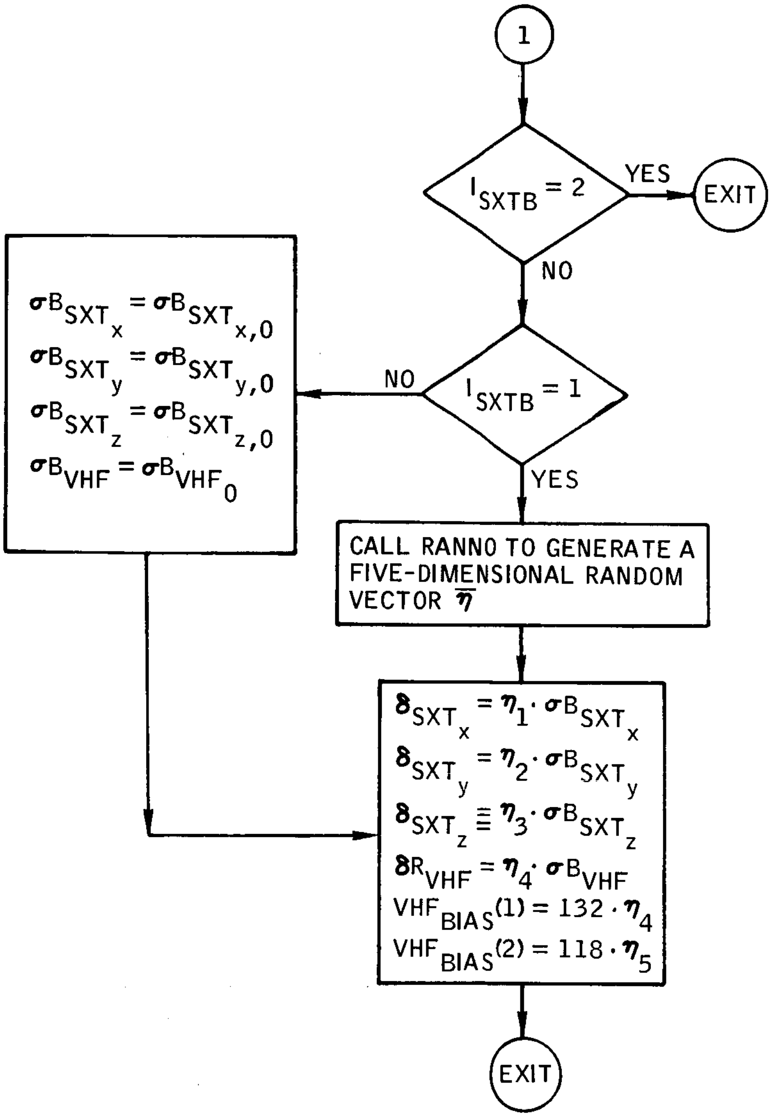
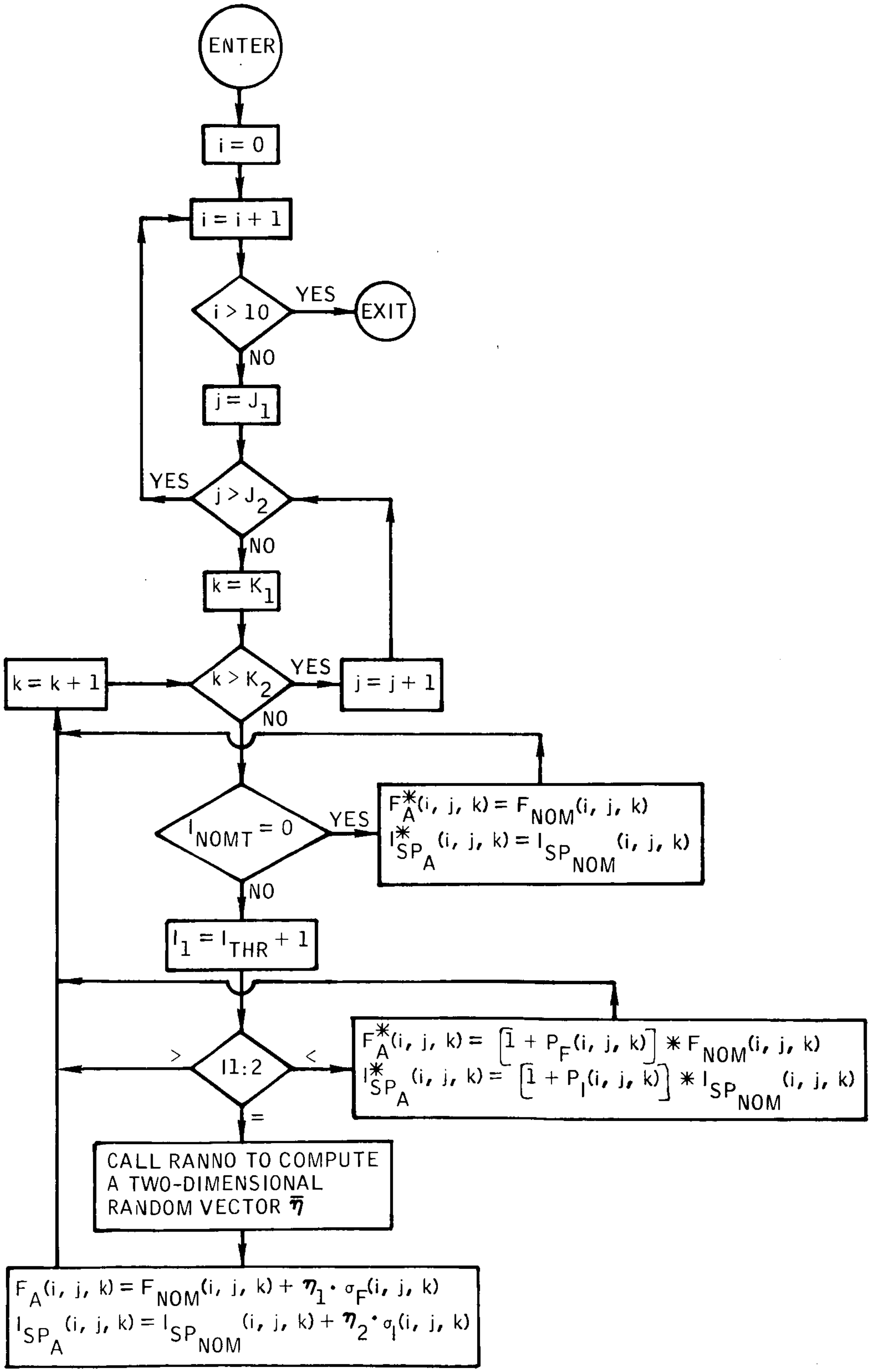

1
$I_{SXTB} = 2$
YES
EXIT
NO
$I_{SXTB} = 1$
NO
$\sigma B_{SXT_x} = \sigma B_{SXT_{x,0}}$
$\sigma B_{SXT_y} = \sigma B_{SXT_{y,0}}$
$\sigma B_{SXT_z} = \sigma B_{SXT_{z,0}}$
$\sigma B_{VHF} = \sigma B_{VHF_0}$
YES
CALL RANNO TO GENERATE A FIVE-DIMENSIONAL RANDOM VECTOR $\overline{\eta}$
$\delta_{SXT_x} = \eta_1 \cdot \sigma B_{SXT_x}$
$\delta_{SXT_y} = \eta_2 \cdot \sigma B_{SXT_y}$
$\delta_{SXT_z} \equiv \eta_3 \cdot \sigma B_{SXT_z}$
$\delta R_{VHF} = \eta_4 \cdot \sigma B_{VHF}$
$VHF_{BIAS}(1) = 132 \cdot \eta_4$
$VHF_{BIAS}(2) = 118 \cdot \eta_5$
EXIT

Figure B-89.- Concluded.

Page 2 of 2


ENTER
$i = 0$
$i = i + 1$
$i > 10$
YES
EXIT
NO
$j = J_1$
YES
$j > J_2$
NO
$k = K_1$
$k = k + 1$
$k > K_2$
YES
$j = j + 1$
NO
$I_{NOMT} = 0$
YES
$F^{*}_{A}(i, j, k) = F_{NOM}(i, j, k)$
$I^{*}_{SP_A}(i, j, k) = I_{SP_{NOM}}(i, j, k)$
NO
$I_1 = I_{THR} + 1$
>
I1:2
<
$F^{*}_{A}(i, j, k) = [1 + P_F(i, j, k)] * F_{NOM}(i, j, k)$
$I^{*}_{SP_A}(i, j, k) = [1 + P_I(i, j, k)] * I_{SP_{NOM}}(i, j, k)$
=
CALL RANNO TO COMPUTE A TWO-DIMENSIONAL RANDOM VECTOR $\overline{\eta}$
$F_A(i, j, k) = F_{NOM}(i, j, k) + \eta_1 \cdot \sigma_F(i, j, k)$
$I_{SP_A}(i, j, k) = I_{SP_{NOM}}(i, j, k) + \eta_2 \cdot \sigma_I(i, j, k)$

Page 1 of 1

Figure B-20. Flow chart of subroutine SETTHR.

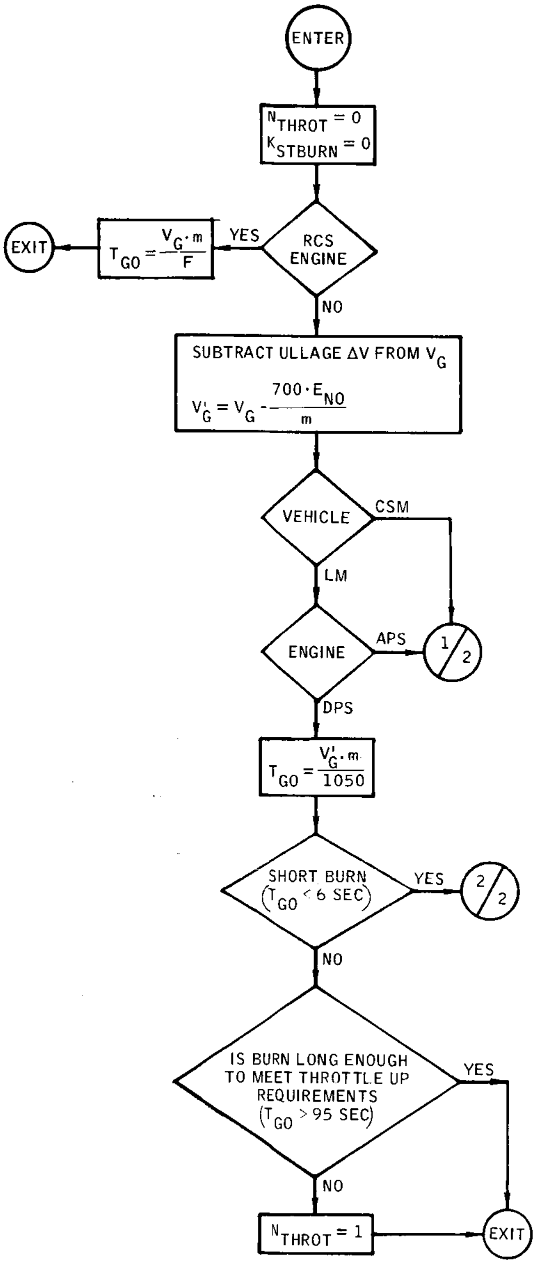
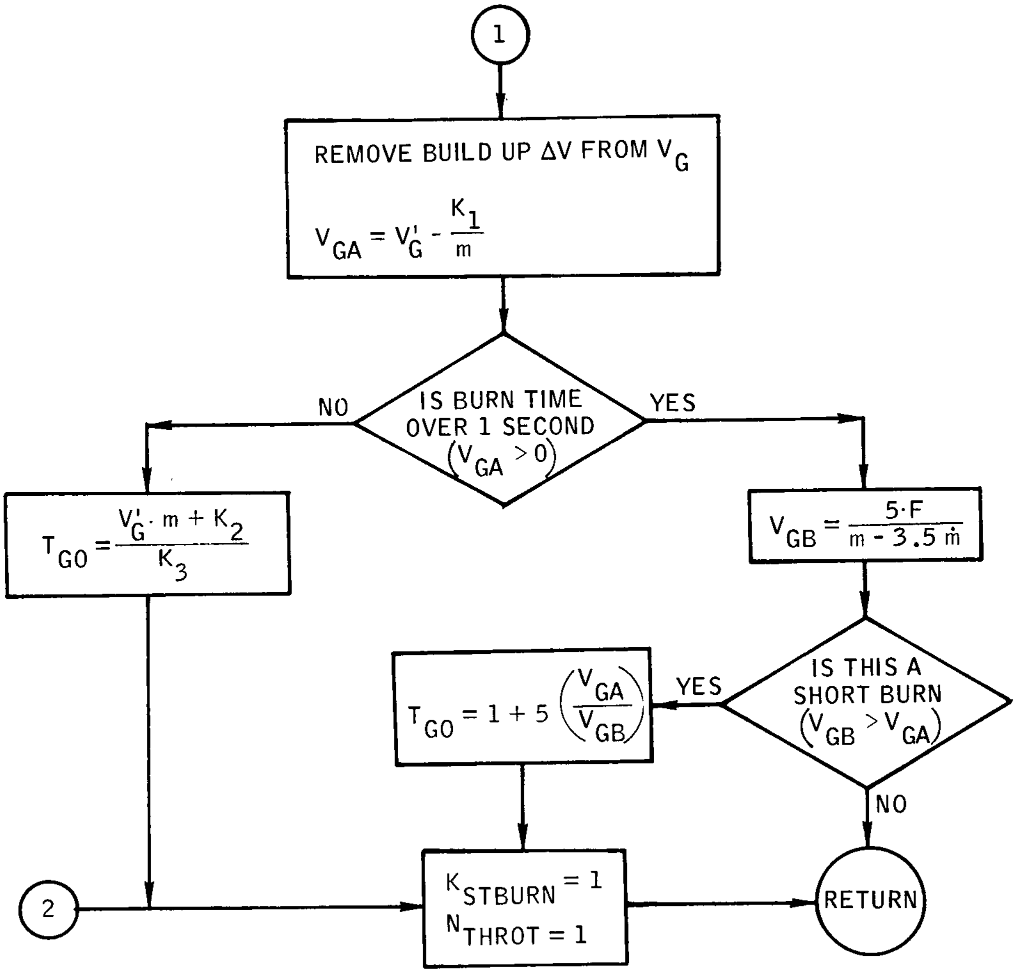

ENTER
$N_{THROT} = 0$
$K_{STBURN} = 0$
RCS ENGINE
YES
$T_{GO} = \frac{V_G \cdot m}{F}$
EXIT
NO
SUBTRACT ULLAGE ΔV FROM V_G
$V_G' = V_G - \frac{700 \cdot E_{NO}}{m}$
VEHICLE
CSM
LM
ENGINE
APS
1/2
DPS
$T_{GO} = \frac{V_G' \cdot m}{1050}$
SHORT BURN ($T_{GO} < 6$ SEC)
YES
2/2
NO
IS BURN LONG ENOUGH TO MEET THROTTLE UP REQUIREMENTS ($T_{GO} > 95$ SEC)
YES
NO
$N_{THROT} = 1$
EXIT

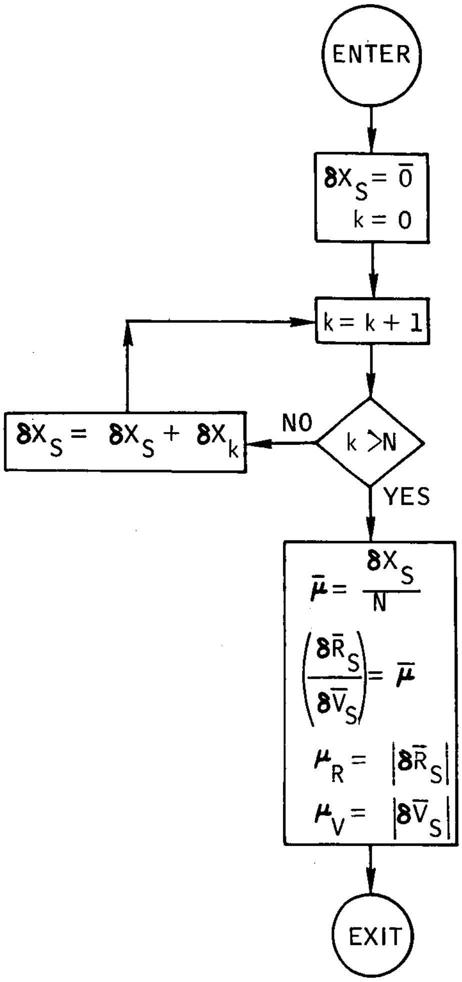
Figure B-91.- Flow chart of subroutine SHORTB.

Page 1 of 2


1
REMOVE BUILD UP ΔV FROM V_G
$V_{GA} = V'_G - \frac{K_1}{m}$
IS BURN TIME OVER 1 SECOND $(V_{GA} > 0)$
NO
YES
$T_{GO} = \frac{V'_G \cdot m + K_2}{K_3}$
$V_{GB} = \frac{5 \cdot F}{m - 3.5 \dot{m}}$
IS THIS A SHORT BURN $(V_{GB} > V_{GA})$
YES
NO
$T_{GO} = 1 + 5 \left(\frac{V_{GA}}{V_{GB}}\right)$
2
$K_{STBURN} = 1$
$N_{THROT} = 1$
RETURN

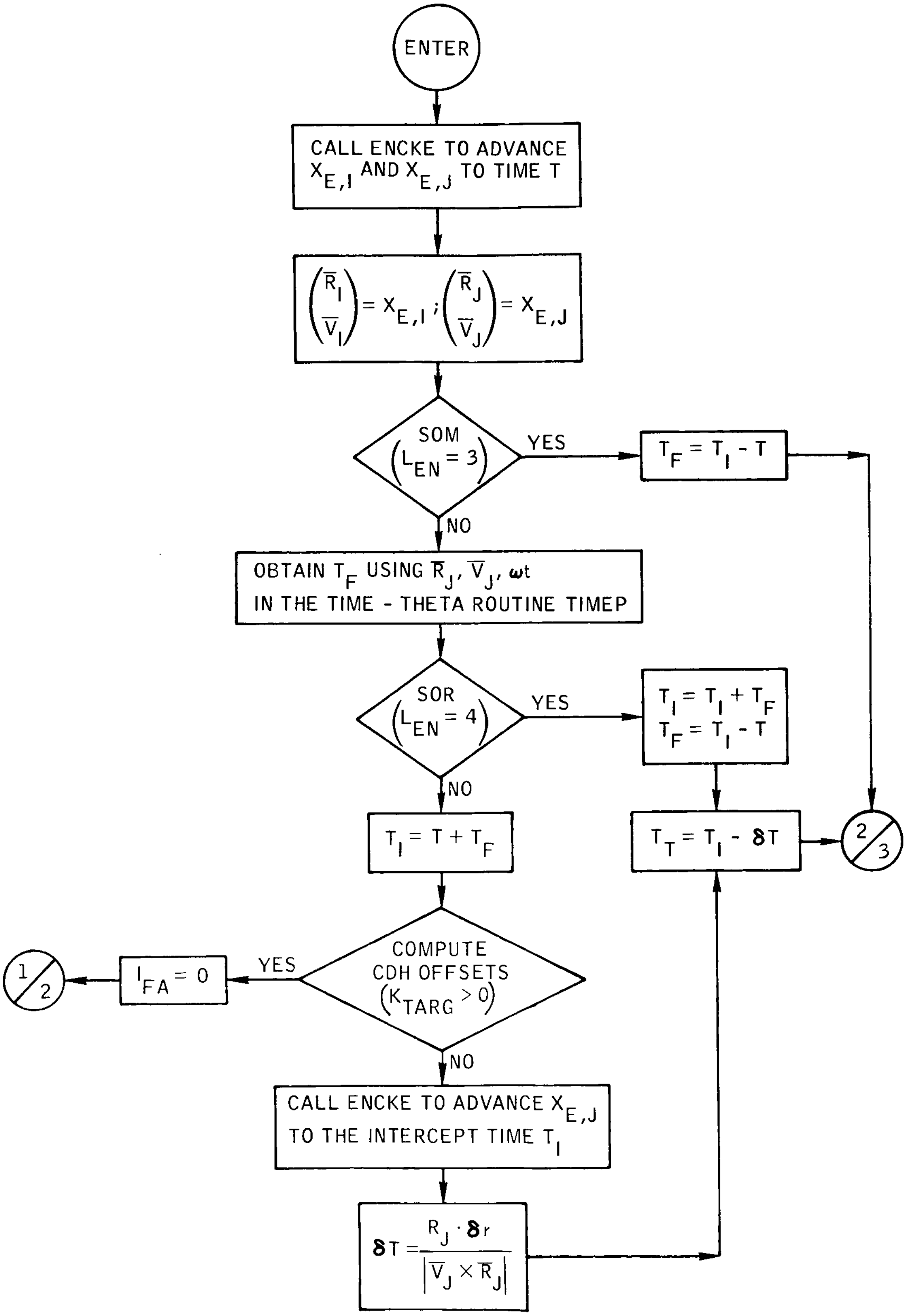
Page 2 of 2

Figure B-91.- Concluded.


ENTER
$\delta X_S = \bar{0}$
$k = 0$
$k = k + 1$
$k > N$
NO
$\delta X_S = \delta X_S + \delta X_k$
YES
$\bar{\mu} = \frac{\delta X_S}{N}$
$\left(\frac{\delta \bar{R}_S}{\delta \bar{V}_S}\right) = \bar{\mu}$
$\mu_R = |\delta \bar{R}_S|$
$\mu_V = |\delta \bar{V}_S|$
EXIT

Page 1 of 1

Figure B-92.- Flow chart of subroutine SMEAN.


ENTER
CALL ENCKE TO ADVANCE $X_{E,I}$ AND $X_{E,J}$ TO TIME T
$\begin{pmatrix}\bar{R}_I\\ \bar{V}_I\end{pmatrix} = X_{E,I}$; $\begin{pmatrix}\bar{R}_J\\ \bar{V}_J\end{pmatrix} = X_{E,J}$
SOM $(L_{EN} = 3)$
YES
NO
$T_F = T_I - T$
OBTAIN T_F USING $\bar{R}_J$, $\bar{V}_J$, ωt IN THE TIME - THETA ROUTINE TIMEP
SOR $(L_{EN} = 4)$
YES
NO
$T_I = T_I + T_F$
$T_F = T_I - T$
$T_I = T + T_F$
$T_T = T_I - \delta T$
2/3
COMPUTE CDH OFFSETS $(K_{TARG} > 0)$
YES
NO
$I_{FA} = 0$
1/2
CALL ENCKE TO ADVANCE $X_{E,J}$ TO THE INTERCEPT TIME T_I
$\delta T = \frac{R_J \cdot \delta r}{|\bar{V}_J \times \bar{R}_J|}$

Page 1 of 3

Figure B-93.- Flow chart of subroutine SOR.

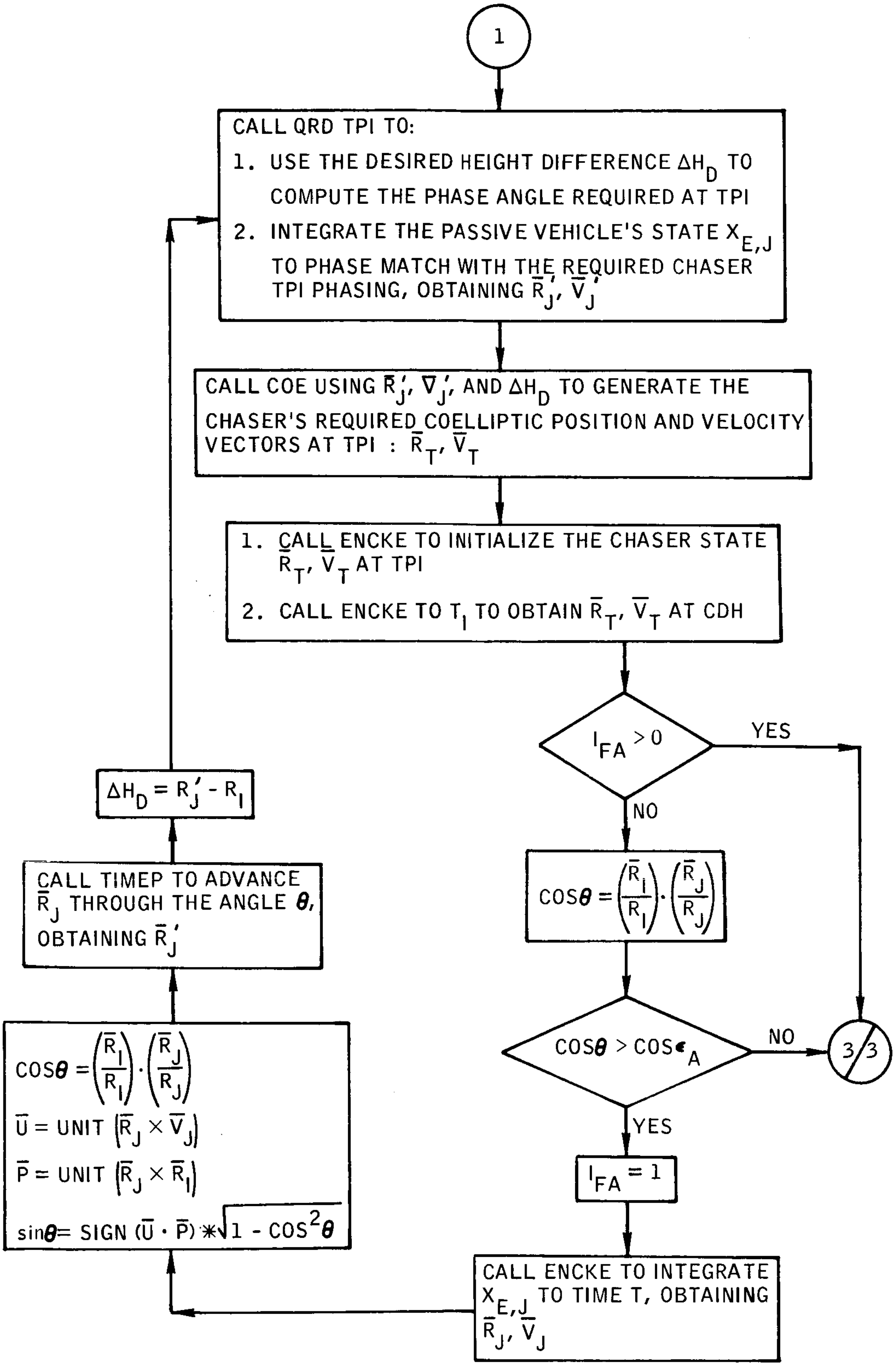
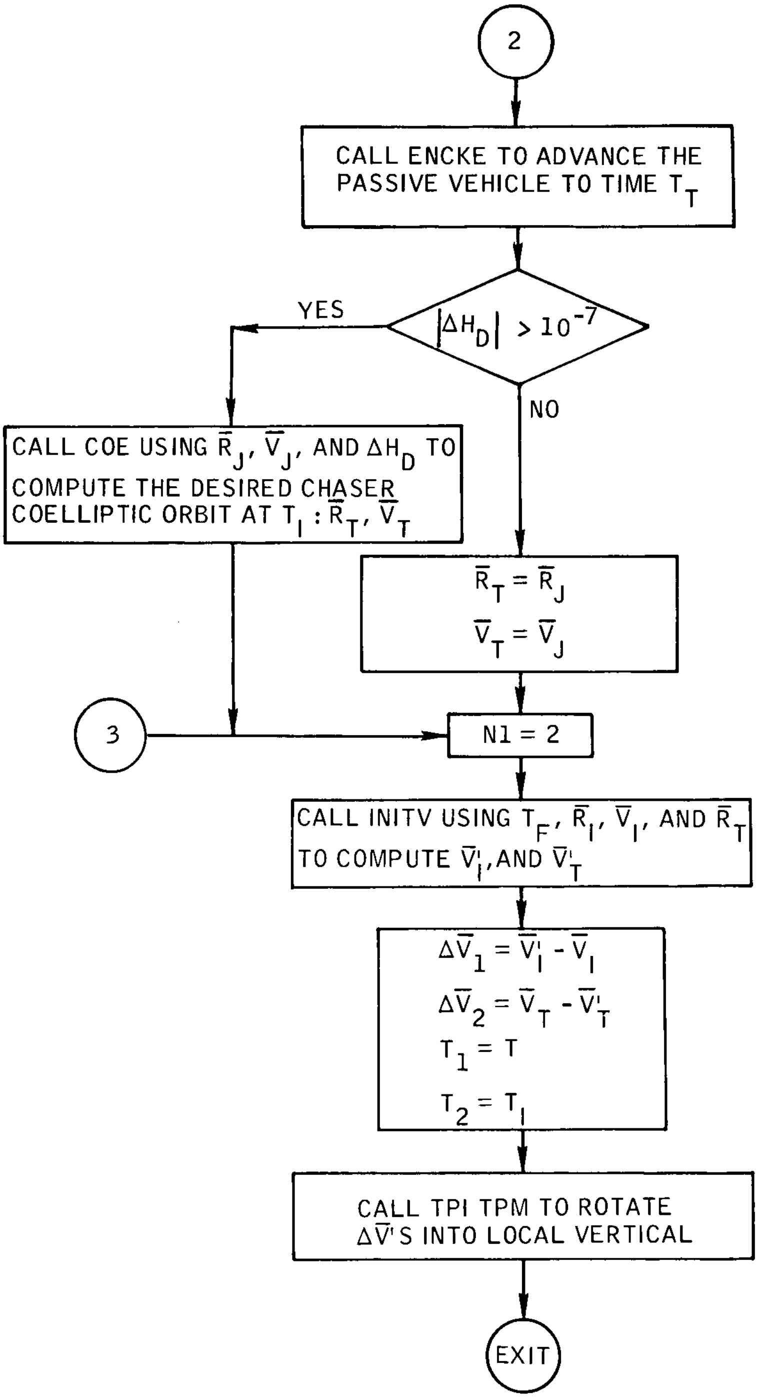

1
CALL QRD TPI TO:
1. USE THE DESIRED HEIGHT DIFFERENCE ΔH_D TO COMPUTE THE PHASE ANGLE REQUIRED AT TPI
2. INTEGRATE THE PASSIVE VEHICLE'S STATE $X_{E,J}$ TO PHASE MATCH WITH THE REQUIRED CHASER TPI PHASING, OBTAINING $\bar{R}_J'$, $\bar{V}_J'$
CALL COE USING $\bar{R}_J'$, $\bar{V}_J'$, AND ΔH_D TO GENERATE THE CHASER'S REQUIRED COELLIPTIC POSITION AND VELOCITY VECTORS AT TPI : $\bar{R}_T$, $\bar{V}_T$
1. CALL ENCKE TO INITIALIZE THE CHASER STATE $\bar{R}_T$, $\bar{V}_T$ AT TPI
2. CALL ENCKE TO T_I TO OBTAIN $\bar{R}_T$, $\bar{V}_T$ AT CDH
$I_{FA} > 0$
YES
NO
$\cos\theta = \left(\frac{\bar{R}_I}{R_I}\right)\cdot\left(\frac{\bar{R}_J}{R_J}\right)$
$\cos\theta > \cos\epsilon_A$
NO
3/3
YES
$I_{FA} = 1$
CALL ENCKE TO INTEGRATE $X_{E,J}$ TO TIME T, OBTAINING $\bar{R}_J$, $\bar{V}_J$
$\cos\theta = \left(\frac{\bar{R}_I}{R_I}\right)\cdot\left(\frac{\bar{R}_J}{R_J}\right)$
$\bar{U} = \text{UNIT}(\bar{R}_J \times \bar{V}_J)$
$\bar{P} = \text{UNIT}(\bar{R}_J \times \bar{R}_I)$
$\sin\theta = \text{SIGN}(\bar{U}\cdot\bar{P}) * \sqrt{1-\cos^2\theta}$
CALL TIMEP TO ADVANCE $\bar{R}_J$ THROUGH THE ANGLE θ, OBTAINING $\bar{R}_J'$
$\Delta H_D = R_J' - R_I$

Figure B-93.- Continued.


2
CALL ENCKE TO ADVANCE THE PASSIVE VEHICLE TO TIME T_T
$|\Delta H_D| > 10^{-7}$
YES
NO
CALL COE USING $\bar{R}_J$, $\bar{V}_J$, AND ΔH_D TO COMPUTE THE DESIRED CHASER COELLIPTIC ORBIT AT T_I : $\bar{R}_T$, $\bar{V}_T$
$\bar{R}_T = \bar{R}_J$
$\bar{V}_T = \bar{V}_J$
3
N1 = 2
CALL INITV USING T_F, $\bar{R}_I$, $\bar{V}_I$, AND $\bar{R}_T$ TO COMPUTE $\bar{V}'_I$, AND $\bar{V}'_T$
$\Delta\bar{V}_1 = \bar{V}'_I - \bar{V}_I$
$\Delta\bar{V}_2 = \bar{V}_T - \bar{V}'_T$
$T_1 = T$
$T_2 = T_I$
CALL TPI TPM TO ROTATE $\Delta\bar{V}$'S INTO LOCAL VERTICAL
EXIT

Page 3 of 3

Figure B-93.- Concluded.

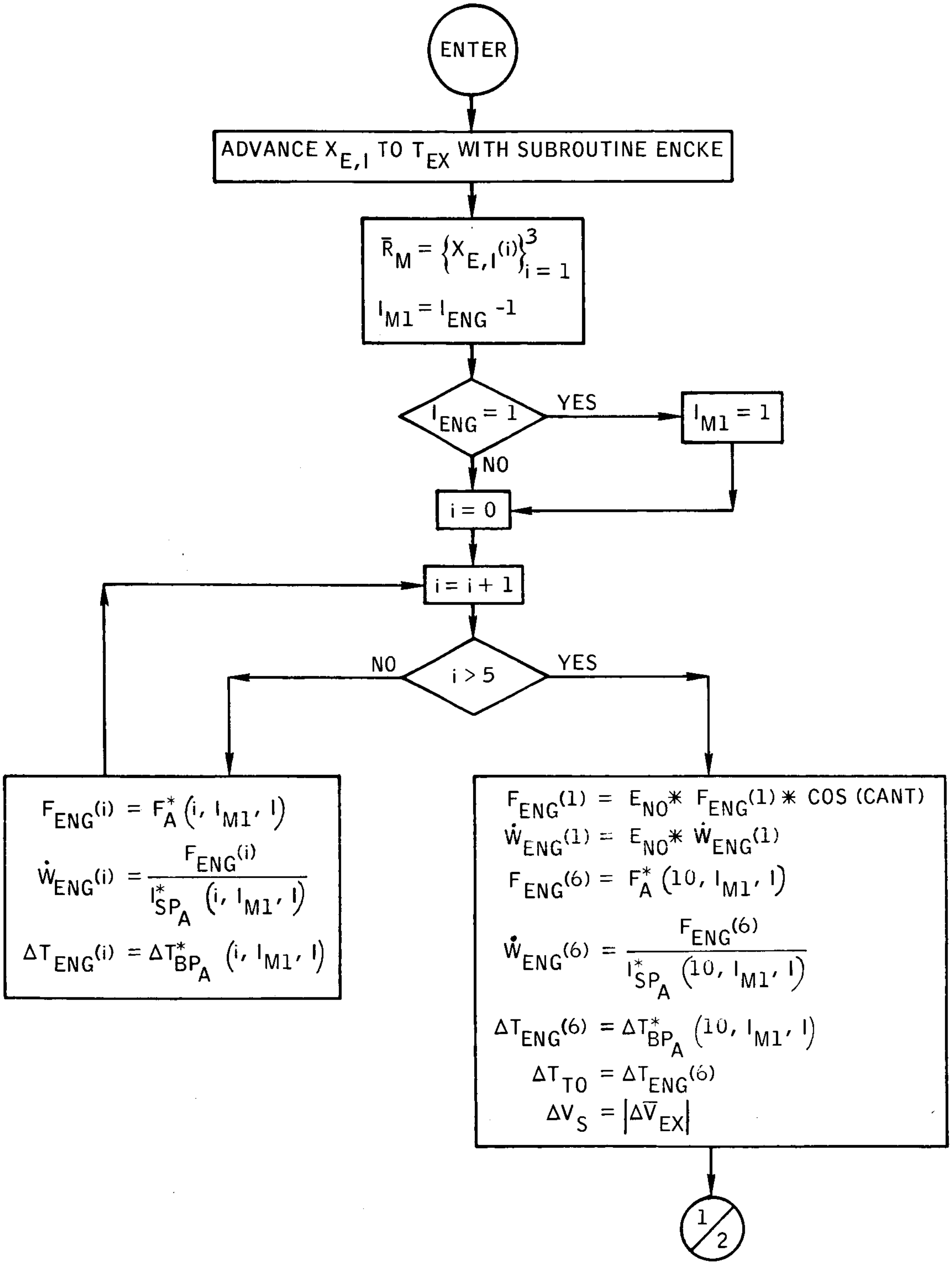
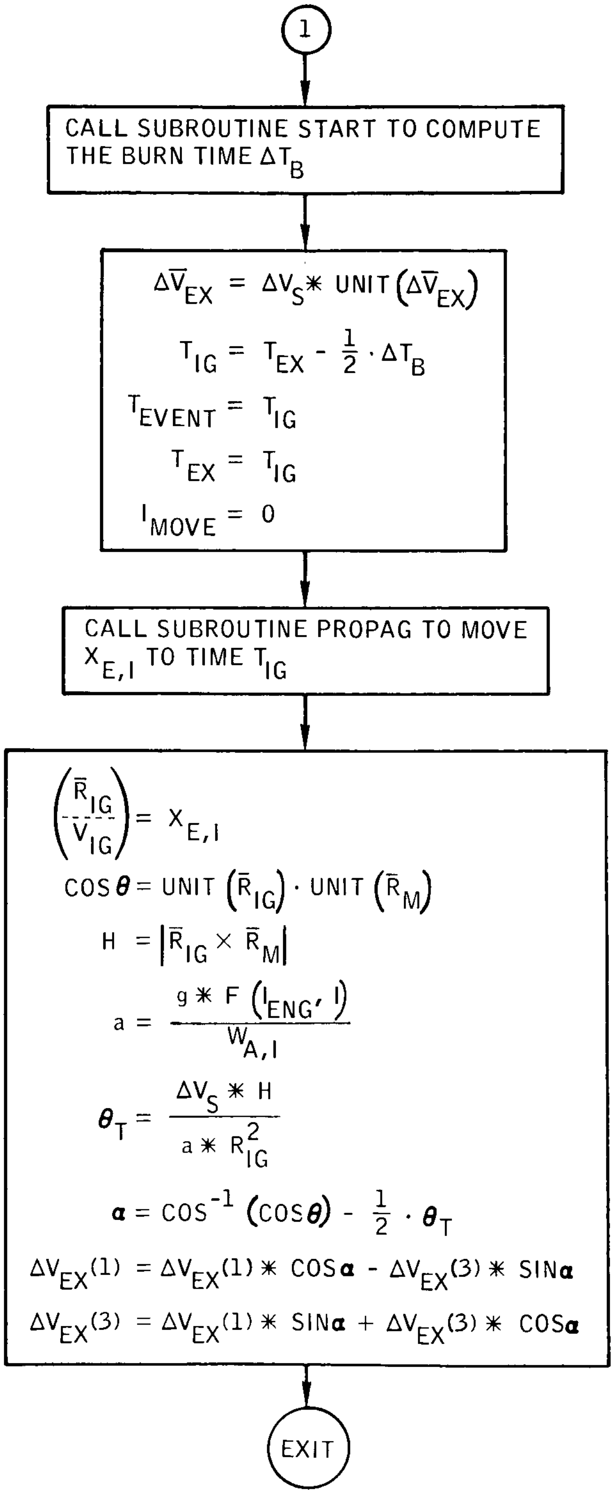

ENTER
ADVANCE $X_{E,I}$ TO T_{EX} WITH SUBROUTINE ENCKE
$\bar{R}_M = \{X_{E,I}(i)\}_{i=1}^{3}$
$I_{M1} = I_{ENG} - 1$
$I_{ENG} = 1$
YES
NO
$I_{M1} = 1$
$i = 0$
$i = i + 1$
$i > 5$
NO
YES
$F_{ENG}(i) = F_A^*(i, I_{M1}, I)$
$\dot{W}_{ENG}(i) = \frac{F_{ENG}(i)}{I_{SP_A}^*(i, I_{M1}, I)}$
$\Delta T_{ENG}(i) = \Delta T_{BP_A}^*(i, I_{M1}, I)$
$F_{ENG}(1) = E_{NO} * F_{ENG}(1) * \text{COS (CANT)}$
$\dot{W}_{ENG}(1) = E_{NO} * \dot{W}_{ENG}(1)$
$F_{ENG}(6) = F_A^*(10, I_{M1}, I)$
$\dot{W}_{ENG}(6) = \frac{F_{ENG}(6)}{I_{SP_A}^*(10, I_{M1}, I)}$
$\Delta T_{ENG}(6) = \Delta T_{BP_A}^*(10, I_{M1}, I)$
$\Delta T_{TO} = \Delta T_{ENG}(6)$
$\Delta V_S = |\Delta \bar{V}_{EX}|$
1/2

Figure B-94.- Flow chart of subroutine SPLIT.

Page 1 of 2



CALL SUBROUTINE START TO COMPUTE THE BURN TIME ΔT_B

$$\Delta\bar{V}_{EX} = \Delta V_S * \text{UNIT}\left(\Delta\bar{V}_{EX}\right)$$

$$T_{IG} = T_{EX} - \frac{1}{2}\cdot\Delta T_B$$

$$T_{EVENT} = T_{IG}$$

$$T_{EX} = T_{IG}$$

$$I_{MOVE} = 0$$

CALL SUBROUTINE PROPAG TO MOVE $X_{E,I}$ TO TIME T_{IG}

$$\begin{pmatrix}\bar{R}_{IG}\\ \hline V_{IG}\end{pmatrix} = X_{E,I}$$

$$\text{COS}\,\theta = \text{UNIT}\left(\bar{R}_{IG}\right)\cdot\text{UNIT}\left(\bar{R}_M\right)$$

$$H = \left|\bar{R}_{IG}\times\bar{R}_M\right|$$

$$a = \frac{g * F\left(I_{ENG}, I\right)}{W_{A,I}}$$

$$\theta_T = \frac{\Delta V_S * H}{a * R_{IG}^2}$$

$$\alpha = \cos^{-1}\left(\cos\theta\right) - \frac{1}{2}\cdot\theta_T$$

$$\Delta V_{EX}(1) = \Delta V_{EX}(1) * \text{COS}\,\alpha - \Delta V_{EX}(3) * \text{SIN}\,\alpha$$

$$\Delta V_{EX}(3) = \Delta V_{EX}(1) * \text{SIN}\,\alpha + \Delta V_{EX}(3) * \text{COS}\,\alpha$$

Page 2 of 2

Figure B-94.- Concluded.

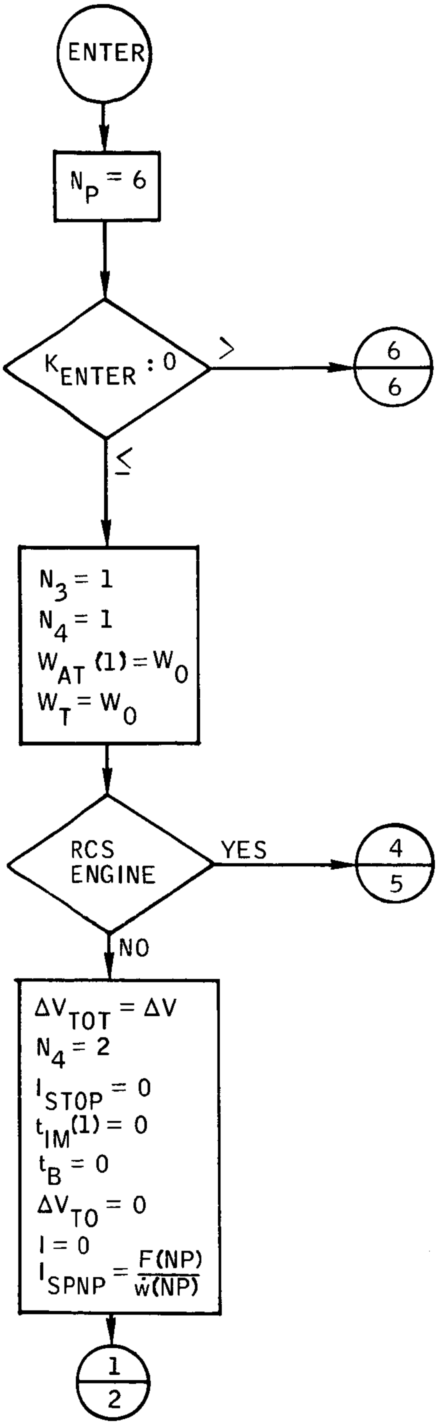


Figure B-95.- Flow chart of subroutine START.

Page 1 of 6

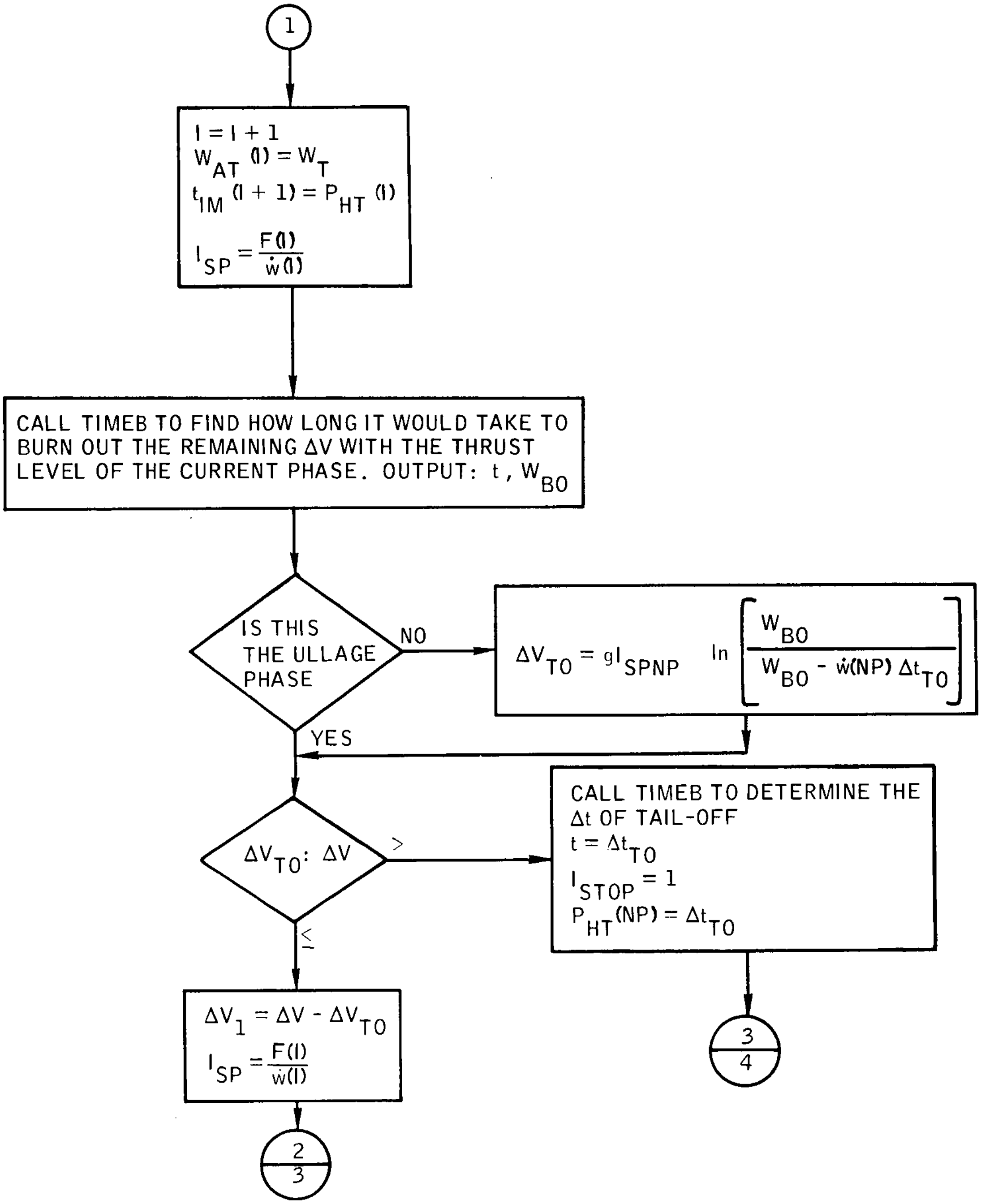

1
$I = I + 1$
$W_{AT}(I) = W_T$
$t_{IM}(I + 1) = P_{HT}(I)$
$I_{SP} = \frac{F(I)}{\dot{w}(I)}$
CALL TIMEB TO FIND HOW LONG IT WOULD TAKE TO BURN OUT THE REMAINING ΔV WITH THE THRUST LEVEL OF THE CURRENT PHASE. OUTPUT: t, W_{BO}
IS THIS THE ULLAGE PHASE
NO
$\Delta V_{TO} = gI_{SPNP} \ln \left[\frac{W_{BO}}{W_{BO} - \dot{w}(NP) \Delta t_{TO}} \right]$
YES
$\Delta V_{TO} : \Delta V$
>
CALL TIMEB TO DETERMINE THE Δt OF TAIL-OFF
$t = \Delta t_{TO}$
$I_{STOP} = 1$
$P_{HT}(NP) = \Delta t_{TO}$
≤
$\Delta V_1 = \Delta V - \Delta V_{TO}$
$I_{SP} = \frac{F(I)}{\dot{w}(I)}$
3/4
2/3

Figure B-95.- Continued.

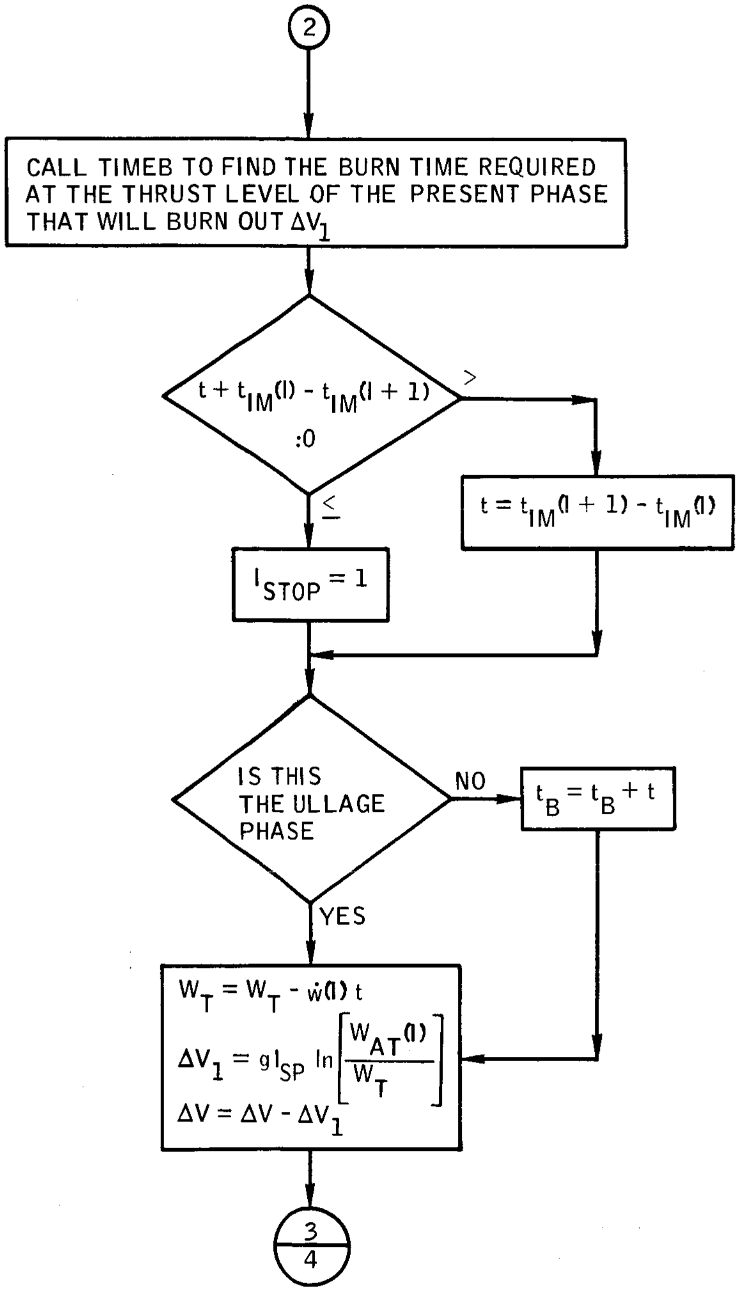
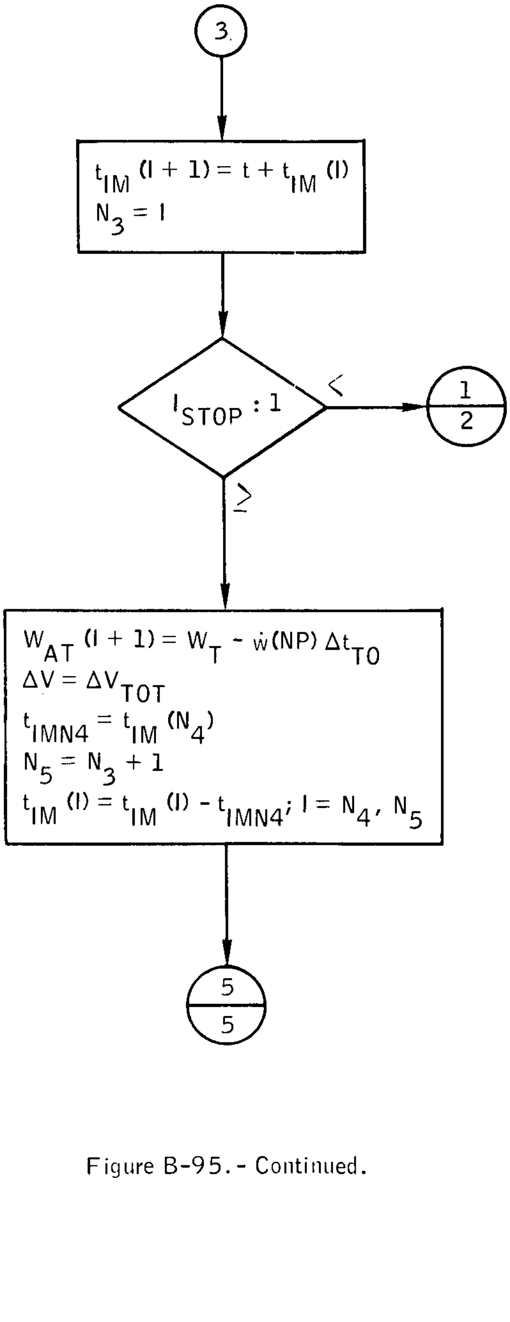

2
CALL TIMEB TO FIND THE BURN TIME REQUIRED AT THE THRUST LEVEL OF THE PRESENT PHASE THAT WILL BURN OUT ΔV_1
$t + t_{IM}(I) - t_{IM}(I+1)$:0
>
$t = t_{IM}(I+1) - t_{IM}(I)$
≤
$I_{STOP} = 1$
IS THIS THE ULLAGE PHASE
NO
$t_B = t_B + t$
YES
$W_T = W_T - \dot{w}(I)\,t$
$\Delta V_1 = g I_{SP} \ln\left[\frac{W_{AT}(I)}{W_T}\right]$
$\Delta V = \Delta V - \Delta V_1$
3
4

Figure B-95.- Continued.


3
$t_{IM}(I + 1) = t + t_{IM}(I)$
$N_3 = I$
$I_{STOP} : 1$
<
1
2
≥
$W_{AT}(I + 1) = W_T - \dot{w}(NP)\,\Delta t_{TO}$
$\Delta V = \Delta V_{TOT}$
$t_{IMN4} = t_{IM}(N_4)$
$N_5 = N_3 + 1$
$t_{IM}(I) = t_{IM}(I) - t_{IMN4};\ I = N_4, N_5$
5
5

Page 4 of 6

Figure B-95.- Continued.

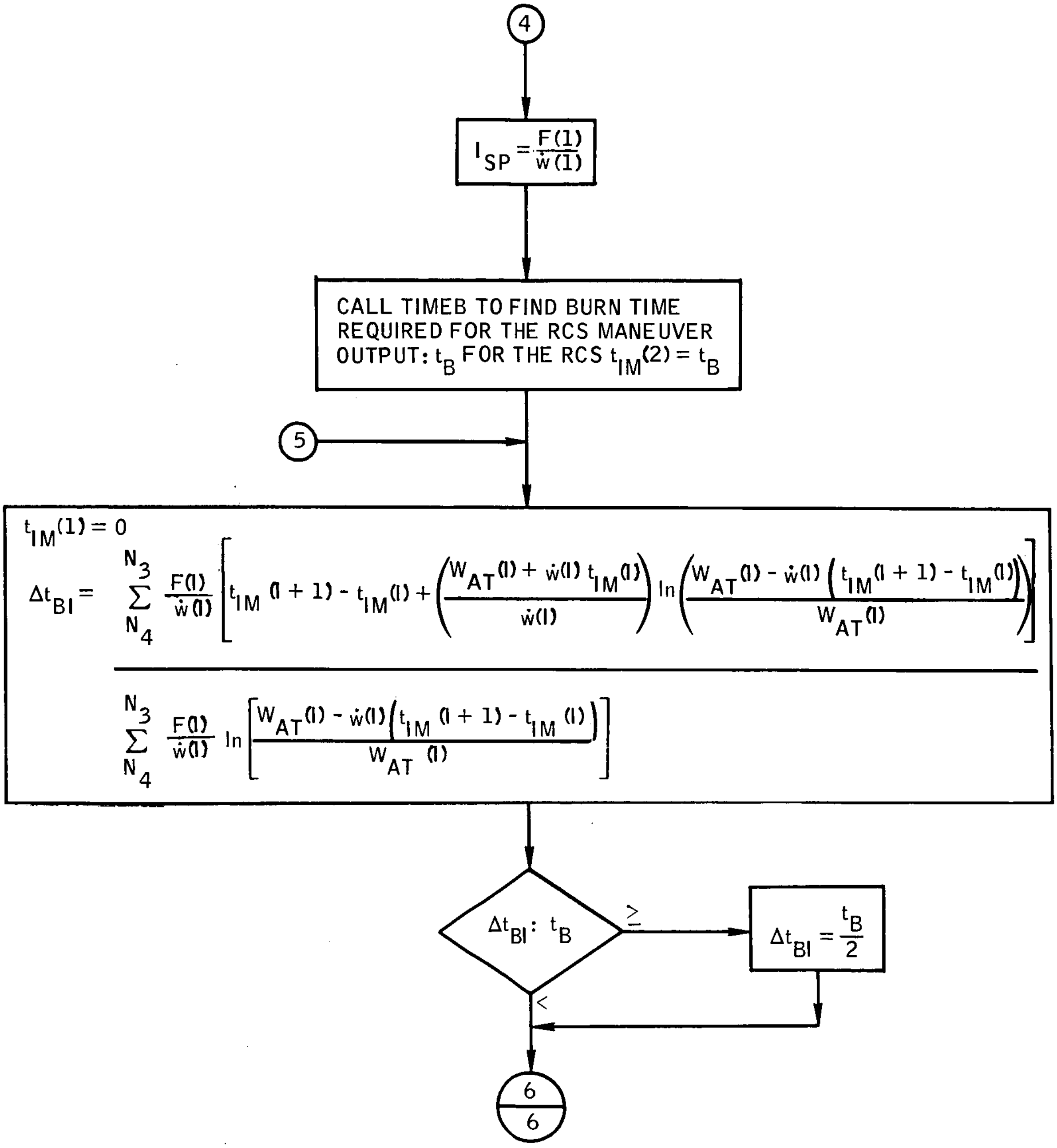

4
$I_{SP} = \frac{F(1)}{\dot{w}(1)}$
CALL TIMEB TO FIND BURN TIME REQUIRED FOR THE RCS MANEUVER
OUTPUT: t_B FOR THE RCS $t_{IM}(2) = t_B$
5
$t_{IM}(1) = 0$
$\Delta t_{BI} = \frac{\sum_{N_4}^{N_3} \frac{F(I)}{\dot{w}(I)} \left[t_{IM}(I+1) - t_{IM}(I) + \left(\frac{W_{AT}(I) + \dot{w}(I)\, t_{IM}(I)}{\dot{w}(I)} \right) \ln \left(\frac{W_{AT}(I) - \dot{w}(I) \left(t_{IM}(I+1) - t_{IM}(I) \right)}{W_{AT}(I)} \right) \right]}{\sum_{N_4}^{N_3} \frac{F(I)}{\dot{w}(I)} \ln \left[\frac{W_{AT}(I) - \dot{w}(I) \left(t_{IM}(I+1) - t_{IM}(I) \right)}{W_{AT}(I)} \right]}$
$\Delta t_{BI} : t_B$
$\geq$
$\Delta t_{BI} = \frac{t_B}{2}$
<
6
6

Figure B-95.- Continued.

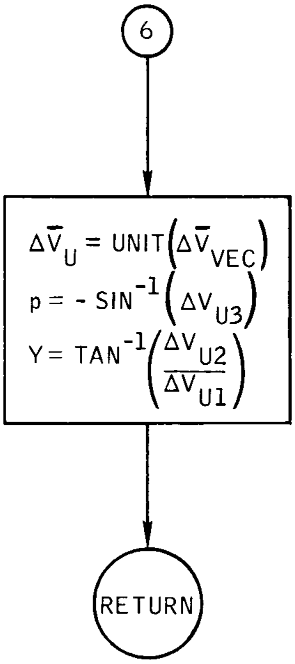
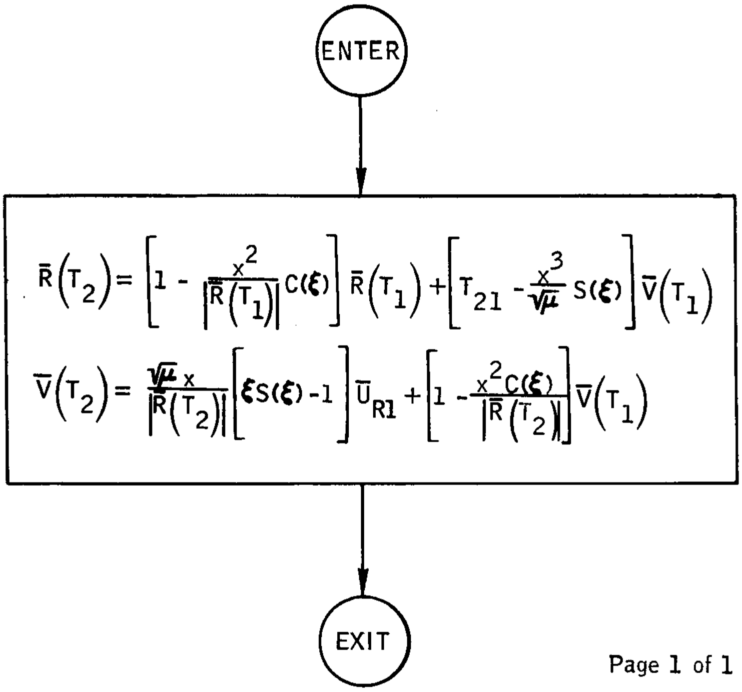

6
$\Delta\bar{V}_U = \text{UNIT}\left(\Delta\bar{V}_{VEC}\right)$
$p = -\text{SIN}^{-1}\left(\Delta V_{U3}\right)$
$Y = \text{TAN}^{-1}\left(\frac{\Delta V_{U2}}{\Delta V_{U1}}\right)$
RETURN

Figure B-95.- Concluded.

Page 6 of 6

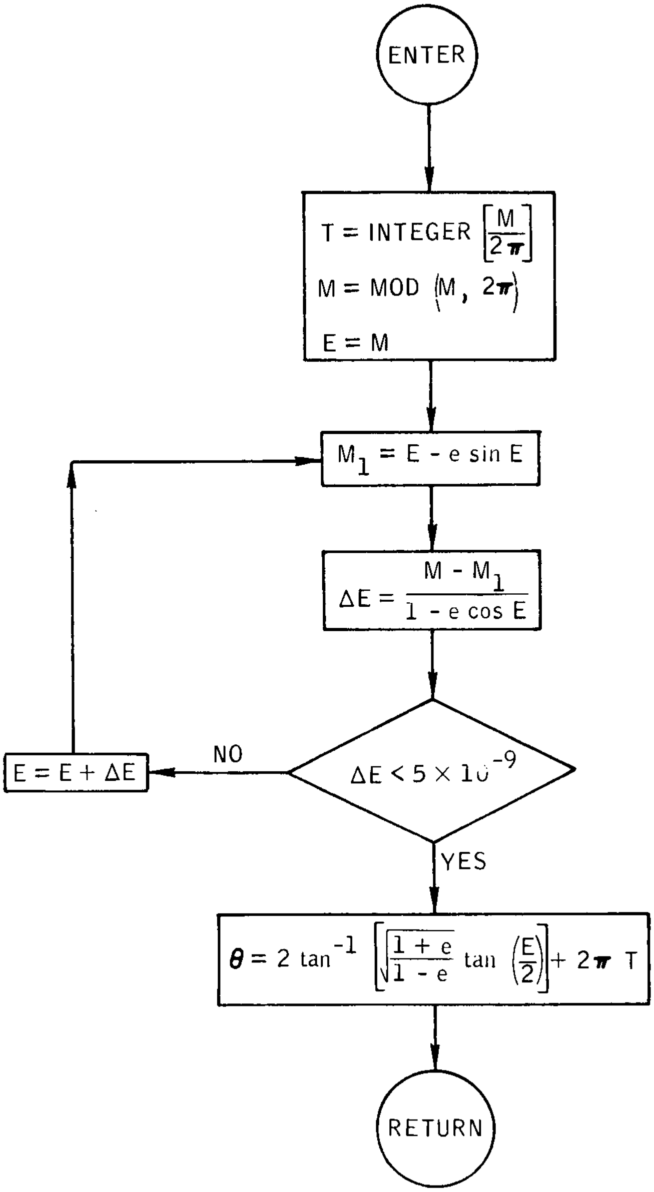


$$\bar{R}\left(T_2\right) = \left[1 - \frac{x^2}{\left|\bar{R}\left(T_1\right)\right|} C(\xi)\right] \bar{R}\left(T_1\right) + \left[T_{21} - \frac{x^3}{\sqrt{\mu}} S(\xi)\right] \bar{V}\left(T_1\right)$$

$$\bar{V}\left(T_2\right) = \frac{\sqrt{\mu}\, x}{\left|\bar{R}\left(T_2\right)\right|}\left[\xi S(\xi) - 1\right] \bar{U}_{R1} + \left[1 - \frac{x^2 C(\xi)}{\left|\bar{R}\left(T_2\right)\right|}\right] \bar{V}\left(T_1\right)$$

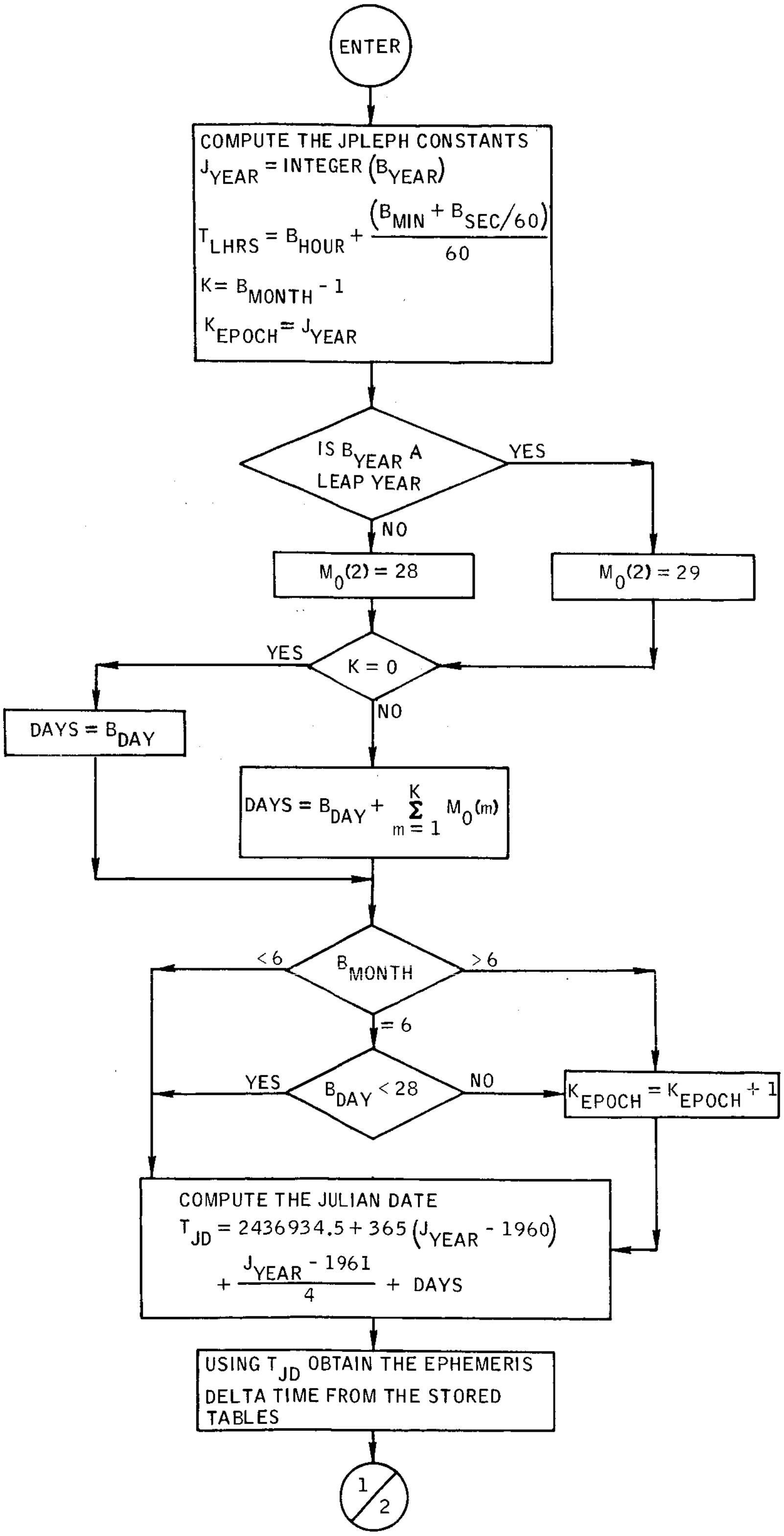
Page 1 of 1

Figure B-96.- Flow chart of subroutine STAYT.


ENTER
$T = \text{INTEGER}\left[\frac{M}{2\pi}\right]$
$M = \text{MOD}\,(M,\ 2\pi)$
$E = M$
$M_1 = E - e \sin E$
$\Delta E = \frac{M - M_1}{1 - e \cos E}$
$\Delta E < 5 \times 10^{-9}$
NO
$E = E + \Delta E$
YES
$\theta = 2 \tan^{-1}\left[\sqrt{\frac{1 + e}{1 - e}} \tan\left(\frac{E}{2}\right)\right] + 2\pi\ T$
RETURN

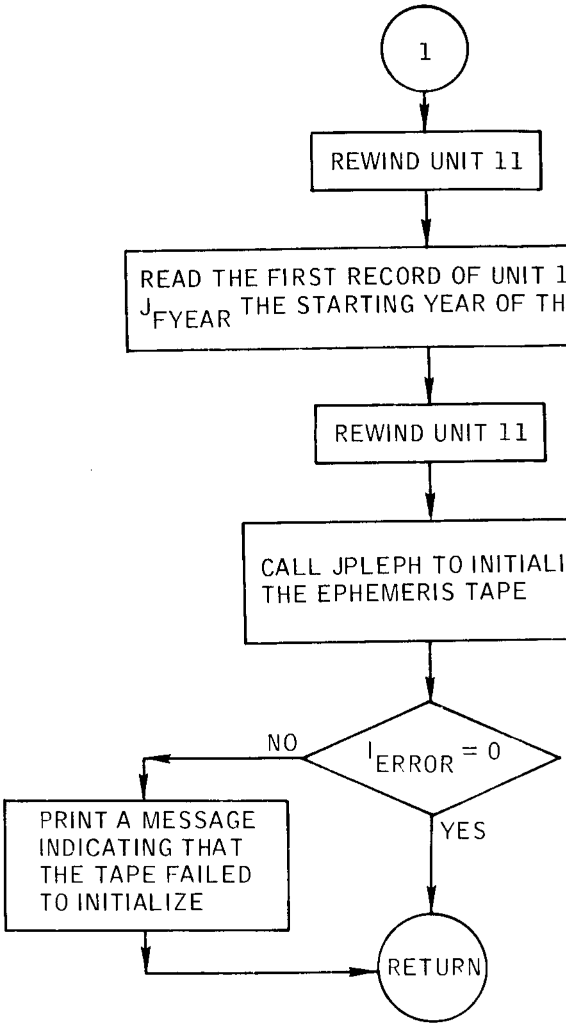
Page 1 of 1

Figure B-97.- Flow chart of function TA.


ENTER
COMPUTE THE JPLEPH CONSTANTS
$J_{YEAR} = \text{INTEGER} (B_{YEAR})$
$T_{LHRS} = B_{HOUR} + \frac{(B_{MIN} + B_{SEC}/60)}{60}$
$K = B_{MONTH} - 1$
$K_{EPOCH} = J_{YEAR}$
IS B_{YEAR} A LEAP YEAR
YES
NO
$M_0(2) = 28$
$M_0(2) = 29$
$K = 0$
YES
NO
$DAYS = B_{DAY}$
$DAYS = B_{DAY} + \sum_{m=1}^{K} M_0(m)$
B_{MONTH}
< 6
> 6
= 6
$B_{DAY} < 28$
YES
NO
$K_{EPOCH} = K_{EPOCH} + 1$
COMPUTE THE JULIAN DATE
$T_{JD} = 2436934.5 + 365 (J_{YEAR} - 1960) + \frac{J_{YEAR} - 1961}{4} + DAYS$
USING T_{JD} OBTAIN THE EPHEMERIS DELTA TIME FROM THE STORED TABLES
1/2

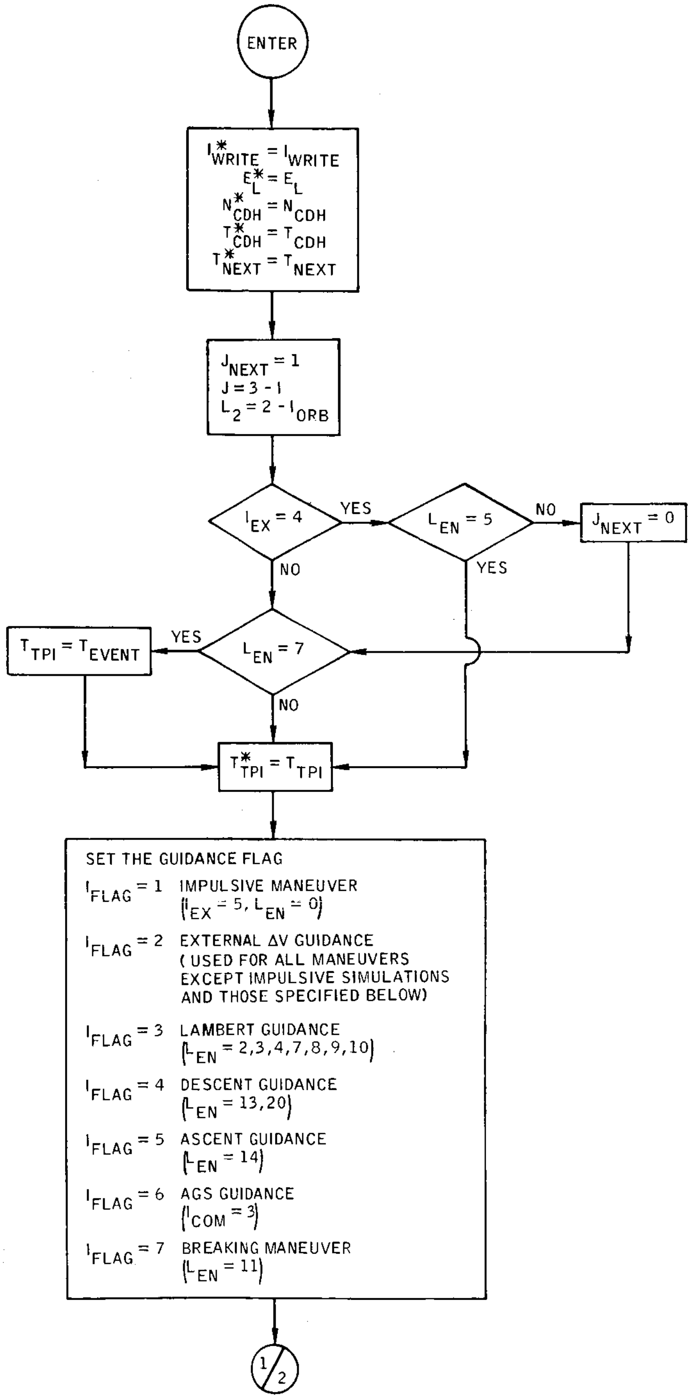
Page 1 of 2

Figure B-98.- Flow chart of subroutine TAPEUP.


1
REWIND UNIT 11
READ THE FIRST RECORD OF UNIT 11 TO OBTAIN J_{FYEAR} THE STARTING YEAR OF THE TAPE
REWIND UNIT 11
CALL JPLEPH TO INITIALIZE THE EPHEMERIS TAPE
$I_{ERROR} = 0$
NO
YES
PRINT A MESSAGE INDICATING THAT THE TAPE FAILED TO INITIALIZE
RETURN

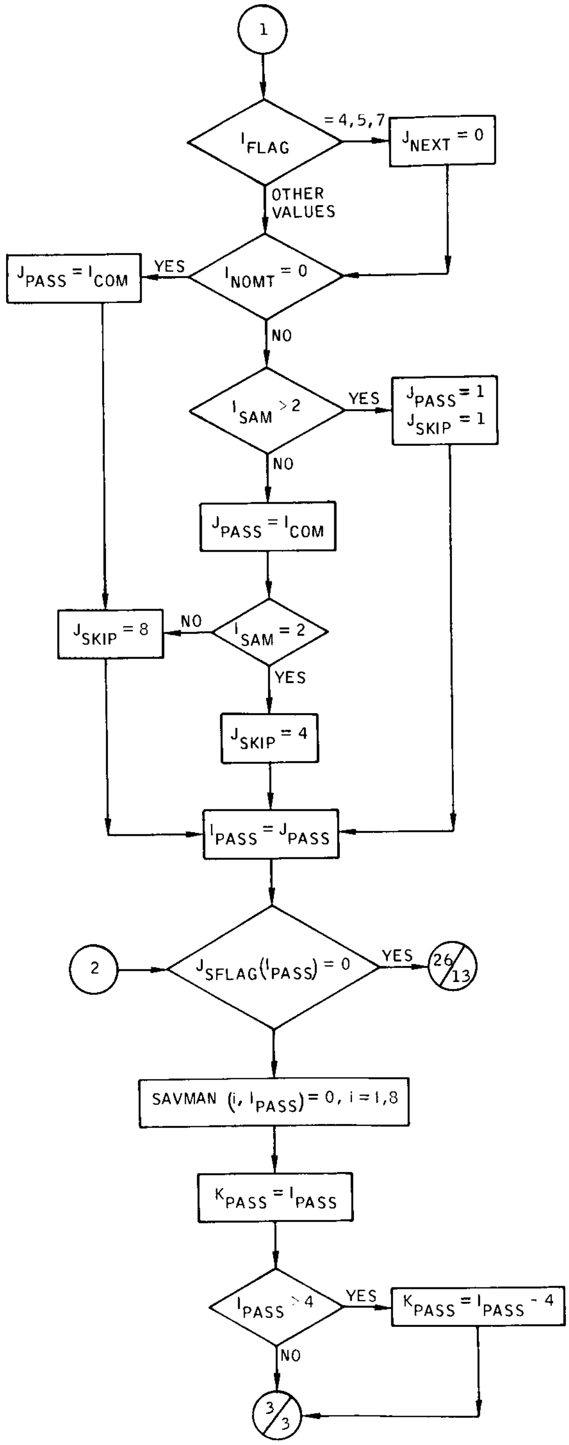
Page 2 of 2

Figure B-98.- Concluded.


ENTER
$I^*_{WRITE} = I_{WRITE}$
$E^*_L = E_L$
$N^*_{CDH} = N_{CDH}$
$T^*_{CDH} = T_{CDH}$
$T^*_{NEXT} = T_{NEXT}$
$J_{NEXT} = 1$
$J = 3 - I$
$L_2 = 2 - I_{ORB}$
$I_{EX} = 4$
YES
$L_{EN} = 5$
NO
$J_{NEXT} = 0$
NO
YES
$T_{TPI} = T_{EVENT}$
YES
$L_{EN} = 7$
NO
$T^*_{TPI} = T_{TPI}$
SET THE GUIDANCE FLAG
$I_{FLAG} = 1$ IMPULSIVE MANEUVER ($I_{EX} = 5$, $L_{EN} = 0$)
$I_{FLAG} = 2$ EXTERNAL ΔV GUIDANCE (USED FOR ALL MANEUVERS EXCEPT IMPULSIVE SIMULATIONS AND THOSE SPECIFIED BELOW)
$I_{FLAG} = 3$ LAMBERT GUIDANCE ($L_{EN} = 2,3,4,7,8,9,10$)
$I_{FLAG} = 4$ DESCENT GUIDANCE ($L_{EN} = 13,20$)
$I_{FLAG} = 5$ ASCENT GUIDANCE ($L_{EN} = 14$)
$I_{FLAG} = 6$ AGS GUIDANCE ($I_{COM} = 3$)
$I_{FLAG} = 7$ BREAKING MANEUVER ($L_{EN} = 11$)
1/2

Page 1 of 14

Figure B-99.- Flow chart of subroutine TARGET.



Page 2 of 14

Figure B-99.- Continued.

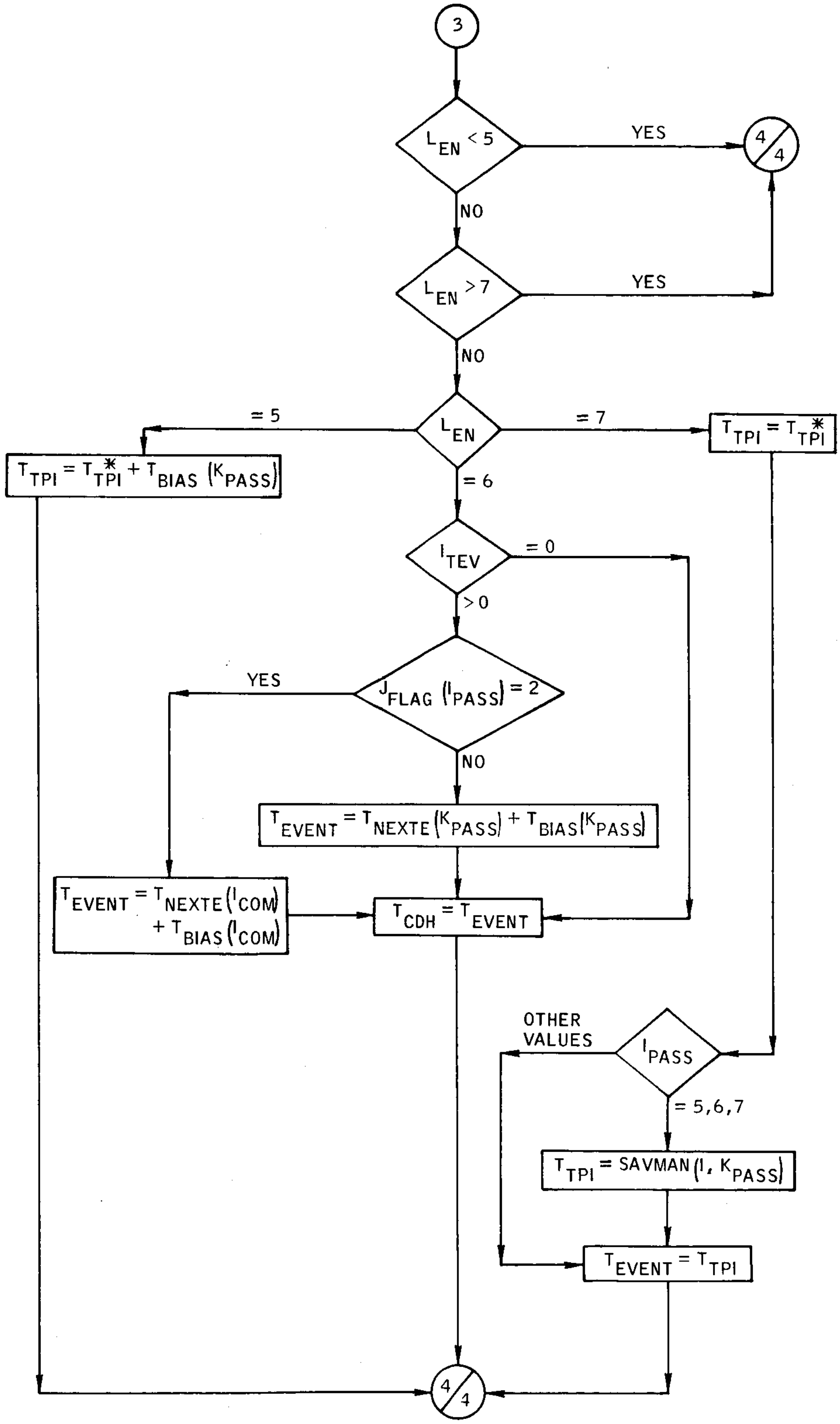


Figure B-99.- Continued.

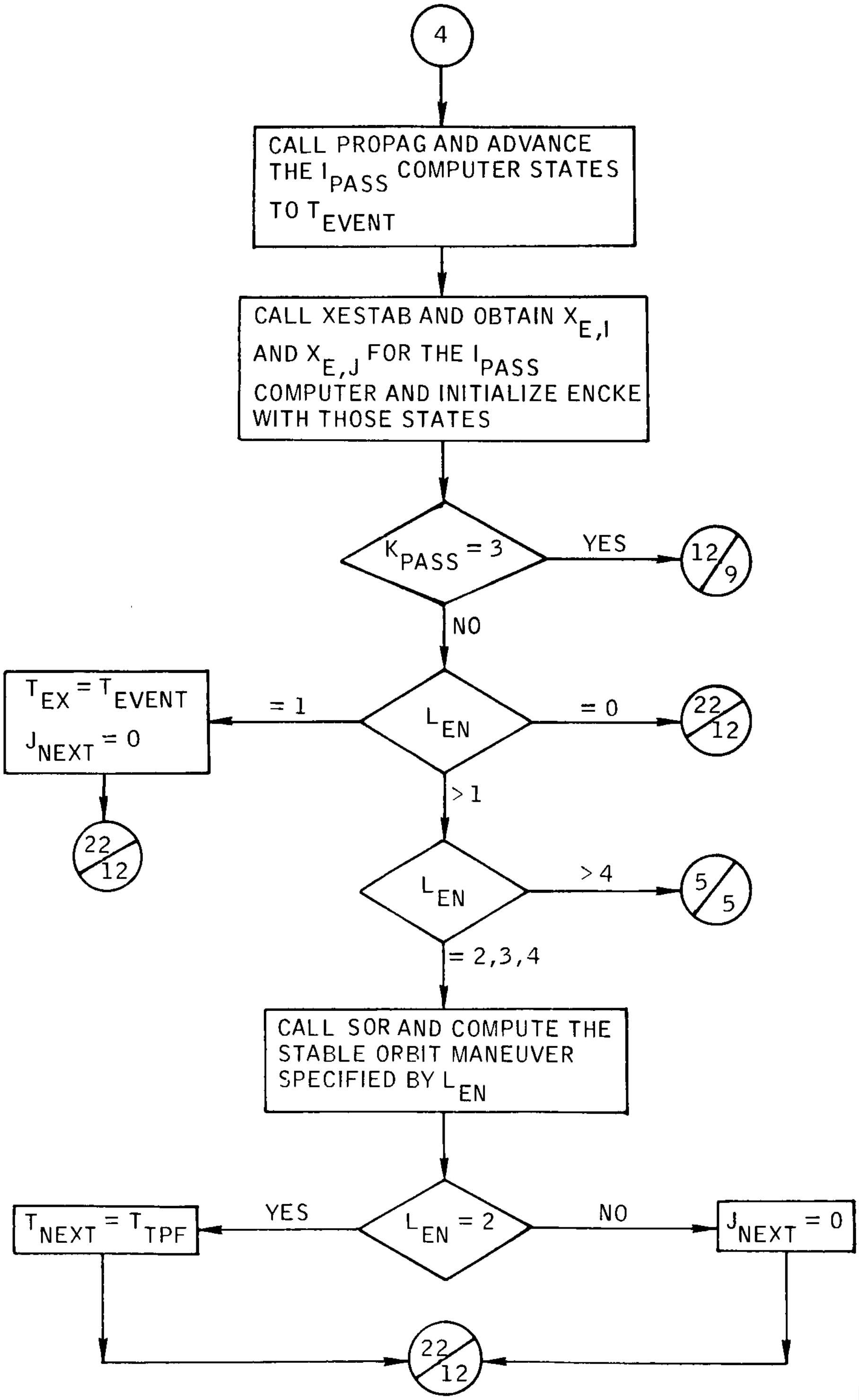

4
CALL PROPAG AND ADVANCE THE I_{PASS} COMPUTER STATES TO T_{EVENT}
CALL XESTAB AND OBTAIN $X_{E,I}$ AND $X_{E,J}$ FOR THE I_{PASS} COMPUTER AND INITIALIZE ENCKE WITH THOSE STATES
$K_{PASS} = 3$
YES
12/9
NO
L_{EN}
= 1
$T_{EX} = T_{EVENT}$
$J_{NEXT} = 0$
22/12
= 0
22/12
> 1
L_{EN}
> 4
5/5
= 2,3,4
CALL SOR AND COMPUTE THE STABLE ORBIT MANEUVER SPECIFIED BY L_{EN}
$L_{EN} = 2$
YES
$T_{NEXT} = T_{TPF}$
NO
$J_{NEXT} = 0$
22/12

Figure B-99.- Continued.

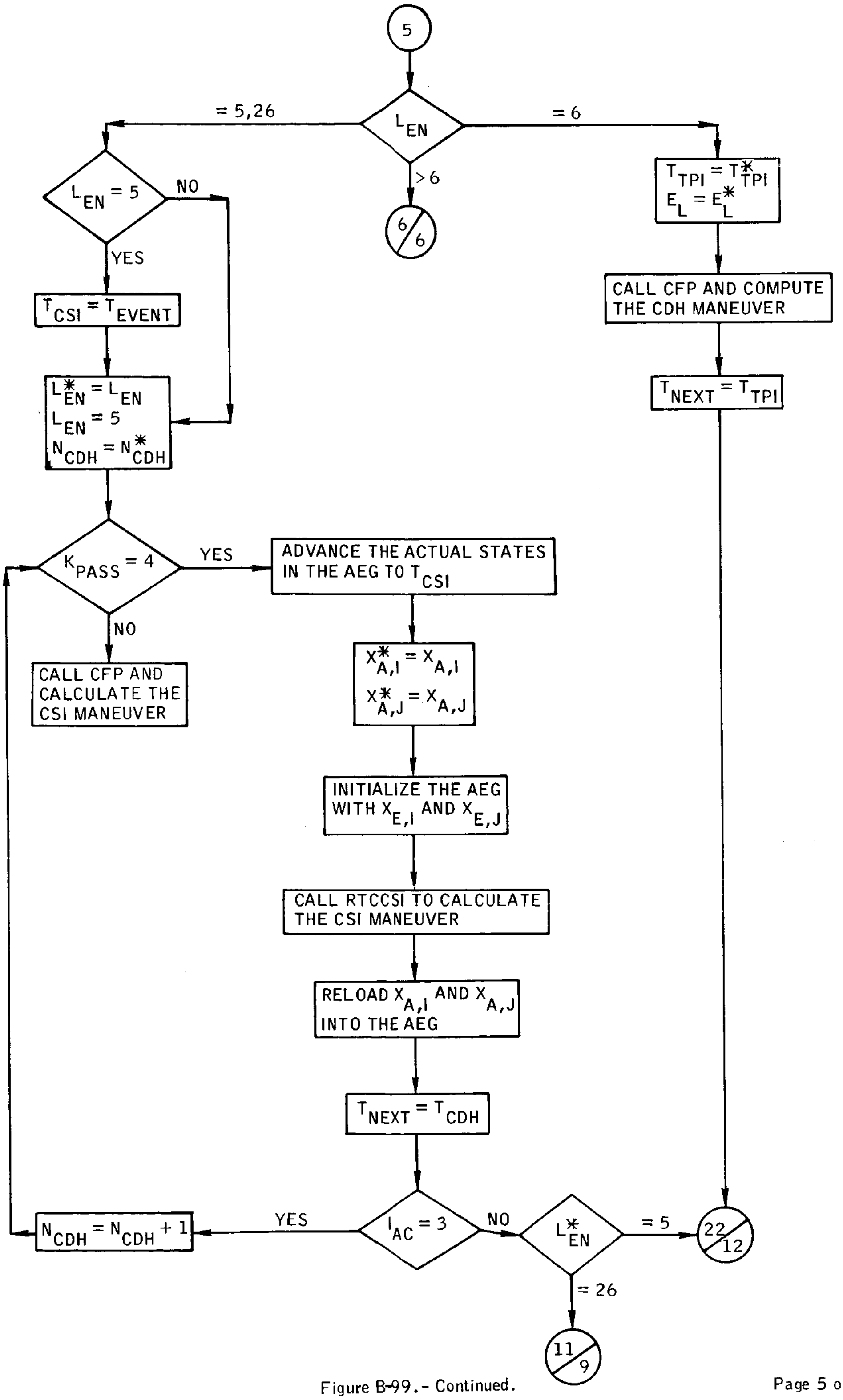
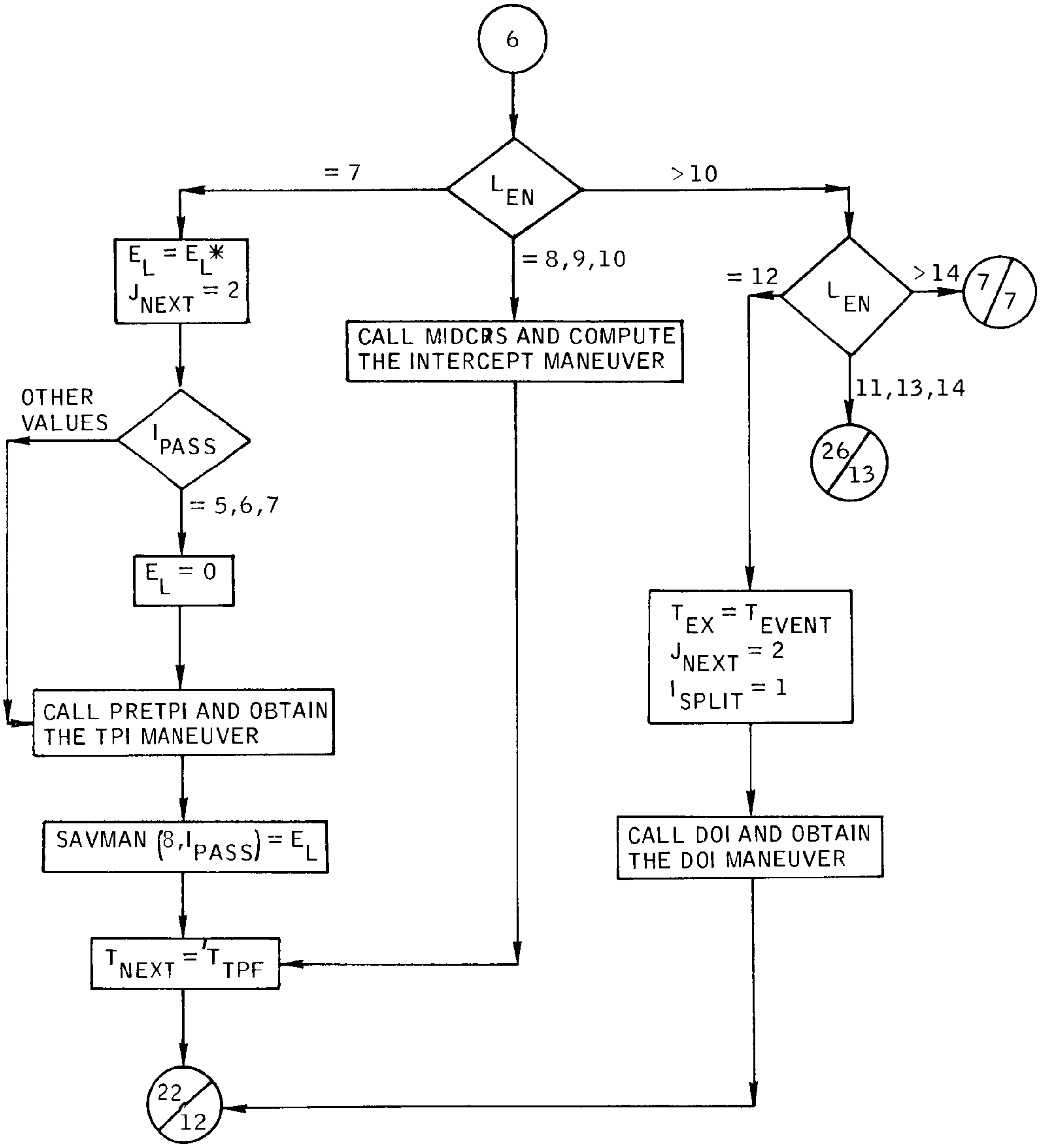
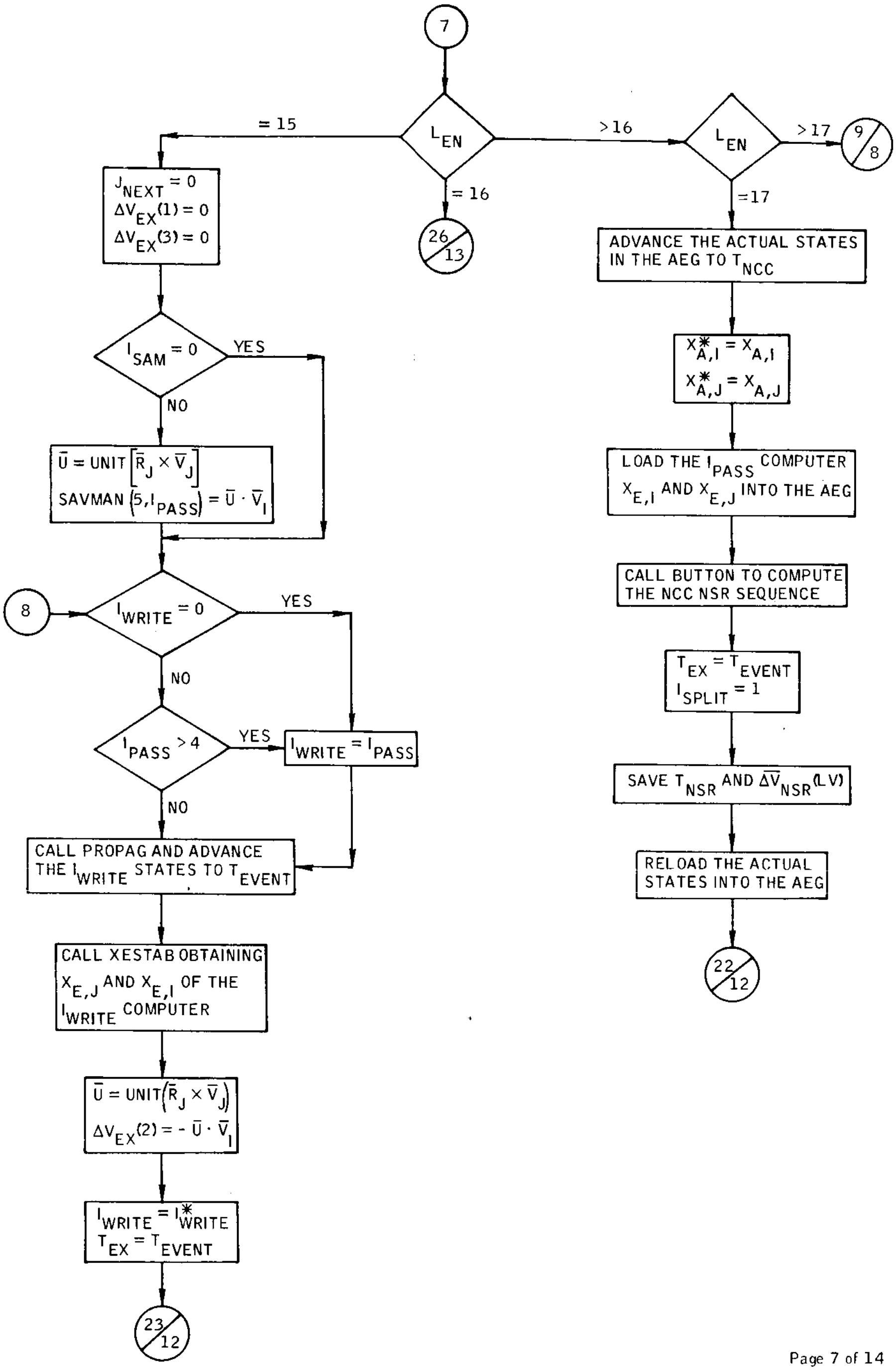


Figure B-99.- Continued.



Page 6 of 14

Figure B-99.- Continued.


7
L_{EN}
= 15
= 16
> 16
26/13
L_{EN}
> 17
9/8
=17
$J_{NEXT} = 0$
$\Delta V_{EX}(1) = 0$
$\Delta V_{EX}(3) = 0$
ADVANCE THE ACTUAL STATES IN THE AEG TO T_{NCC}
$I_{SAM} = 0$
YES
NO
$X^*_{A,I} = X_{A,I}$
$X^*_{A,J} = X_{A,J}$
$\bar{U} = \text{UNIT}\left[\bar{R}_J \times \bar{V}_J\right]$
$\text{SAVMAN}\left(5, I_{PASS}\right) = \bar{U} \cdot \bar{V}_I$
LOAD THE I_{PASS} COMPUTER $X_{E,I}$ AND $X_{E,J}$ INTO THE AEG
CALL BUTTON TO COMPUTE THE NCC NSR SEQUENCE
8
$I_{WRITE} = 0$
YES
NO
$T_{EX} = T_{EVENT}$
$I_{SPLIT} = 1$
$I_{PASS} > 4$
YES
$I_{WRITE} = I_{PASS}$
NO
SAVE T_{NSR} AND $\Delta\bar{V}_{NSR}$(LV)
CALL PROPAG AND ADVANCE THE I_{WRITE} STATES TO T_{EVENT}
RELOAD THE ACTUAL STATES INTO THE AEG
22/12
CALL XESTAB OBTAINING $X_{E,J}$ AND $X_{E,I}$ OF THE I_{WRITE} COMPUTER
$\bar{U} = \text{UNIT}\left(\bar{R}_J \times \bar{V}_J\right)$
$\Delta V_{EX}(2) = - \bar{U} \cdot \bar{V}_I$
$I_{WRITE} = I^*_{WRITE}$
$T_{EX} = T_{EVENT}$
23/12

Page 7 of 14

Figure B-99.- Continued.

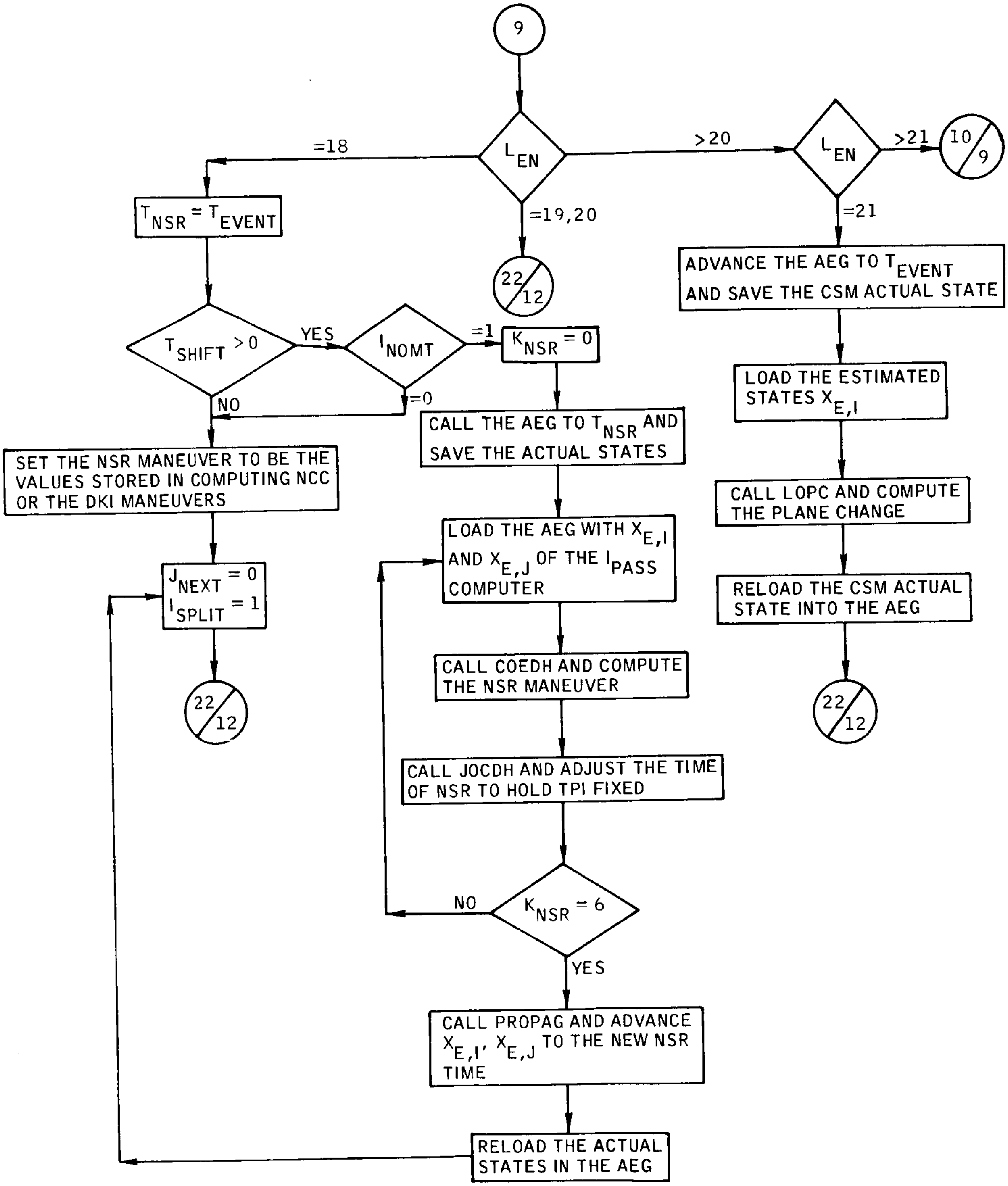
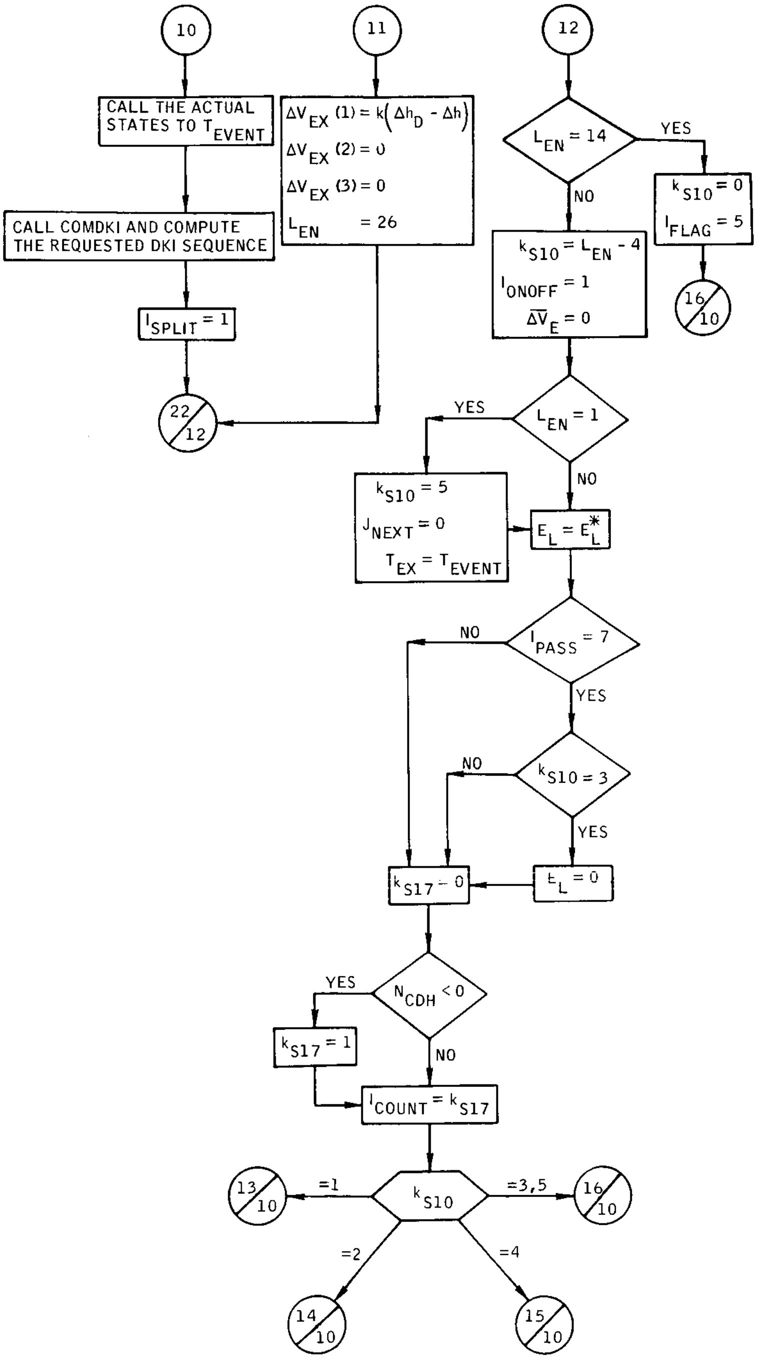

9
L_{EN}
=18
>20
=19,20
L_{EN}
>21
10/9
=21
$T_{NSR} = T_{EVENT}$
22/12
ADVANCE THE AEG TO T_{EVENT} AND SAVE THE CSM ACTUAL STATE
$T_{SHIFT} > 0$
YES
I_{NOMT}
=1
$K_{NSR} = 0$
=0
NO
LOAD THE ESTIMATED STATES $X_{E,I}$
CALL THE AEG TO T_{NSR} AND SAVE THE ACTUAL STATES
SET THE NSR MANEUVER TO BE THE VALUES STORED IN COMPUTING NCC OR THE DKI MANEUVERS
CALL LOPC AND COMPUTE THE PLANE CHANGE
LOAD THE AEG WITH $X_{E,I}$ AND $X_{E,J}$ OF THE I_{PASS} COMPUTER
$J_{NEXT} = 0$
$I_{SPLIT} = 1$
RELOAD THE CSM ACTUAL STATE INTO THE AEG
CALL COEDH AND COMPUTE THE NSR MANEUVER
22/12
22/12
CALL JOCDH AND ADJUST THE TIME OF NSR TO HOLD TPI FIXED
NO
$K_{NSR} = 6$
YES
CALL PROPAG AND ADVANCE $X_{E,I}$, $X_{E,J}$ TO THE NEW NSR TIME
RELOAD THE ACTUAL STATES IN THE AEG

Figure B-99.- Continued.



Page 9 of 14

Figure B-99.- Continued.

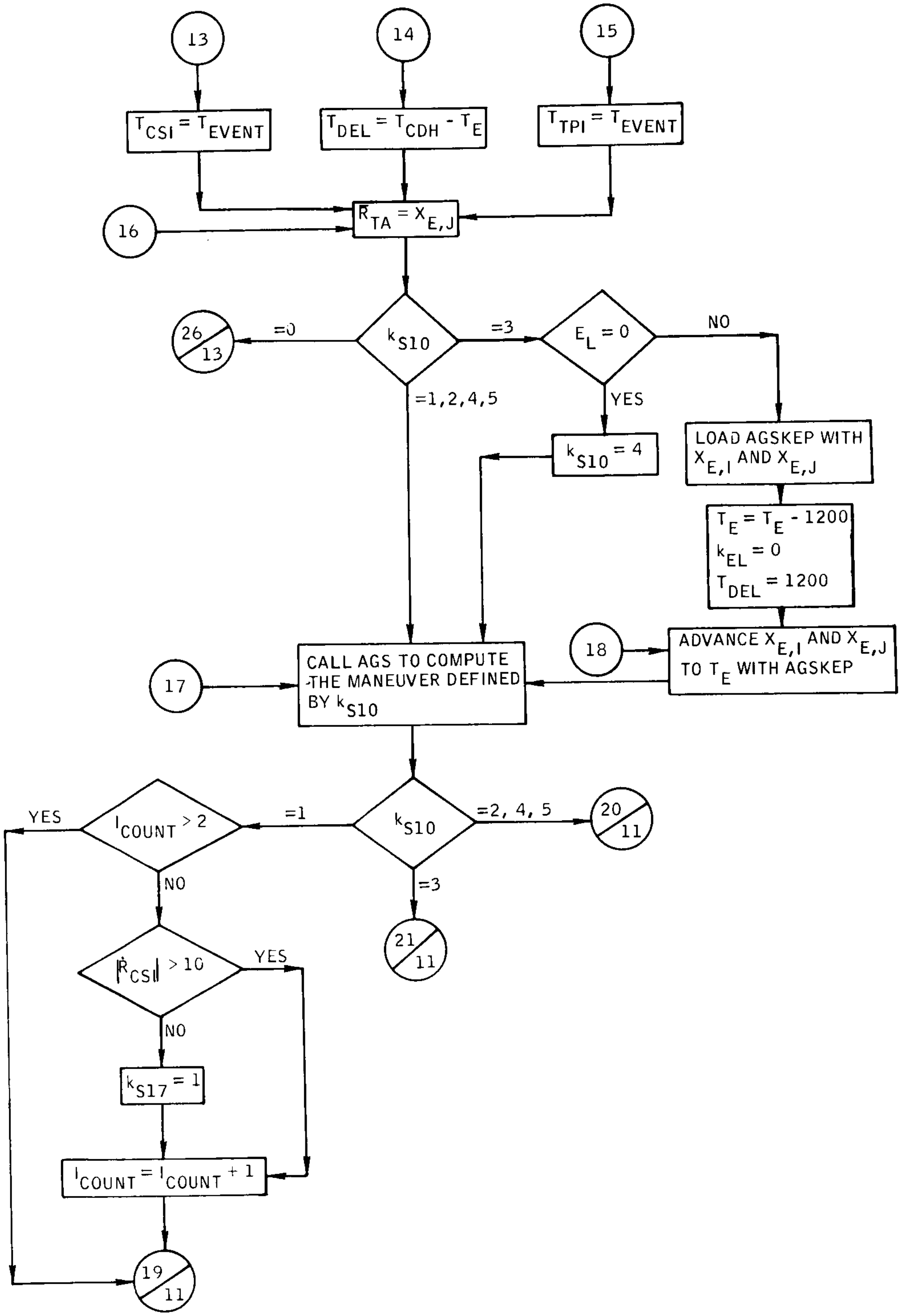


Figure B-99.- Continued.

Page 10 of 14

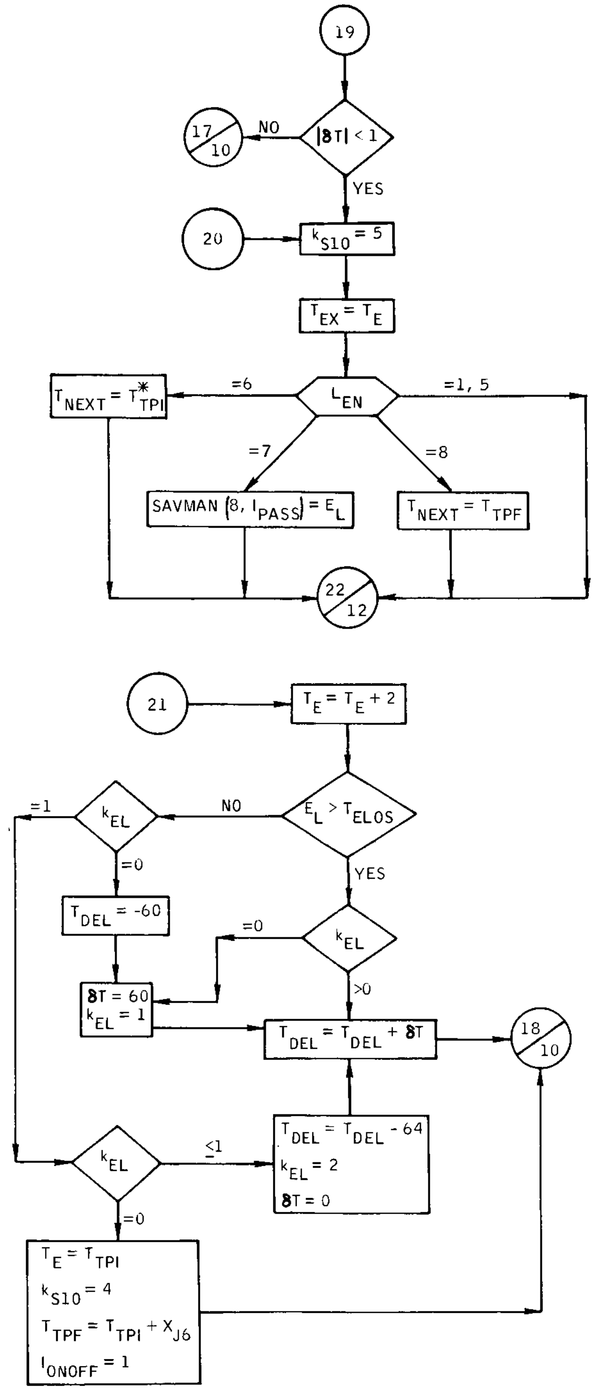


Figure B-99.- Continued.

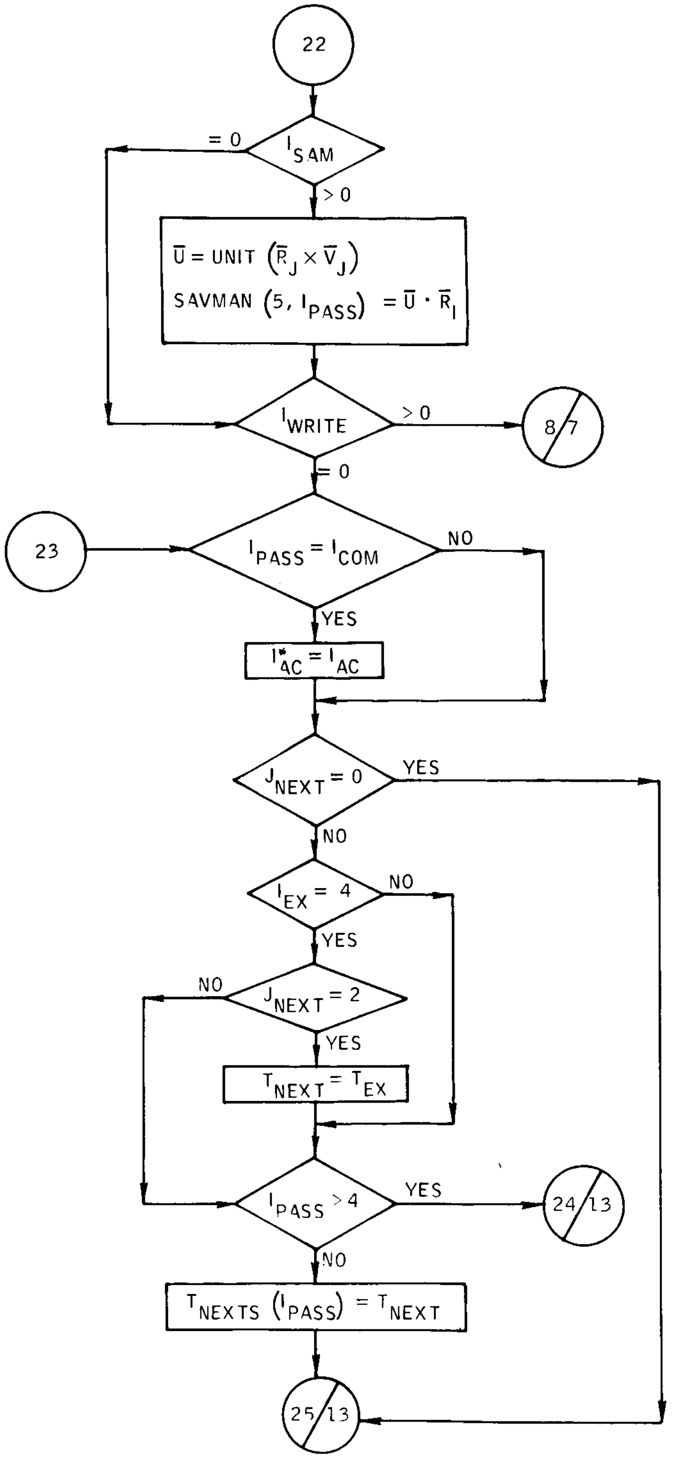
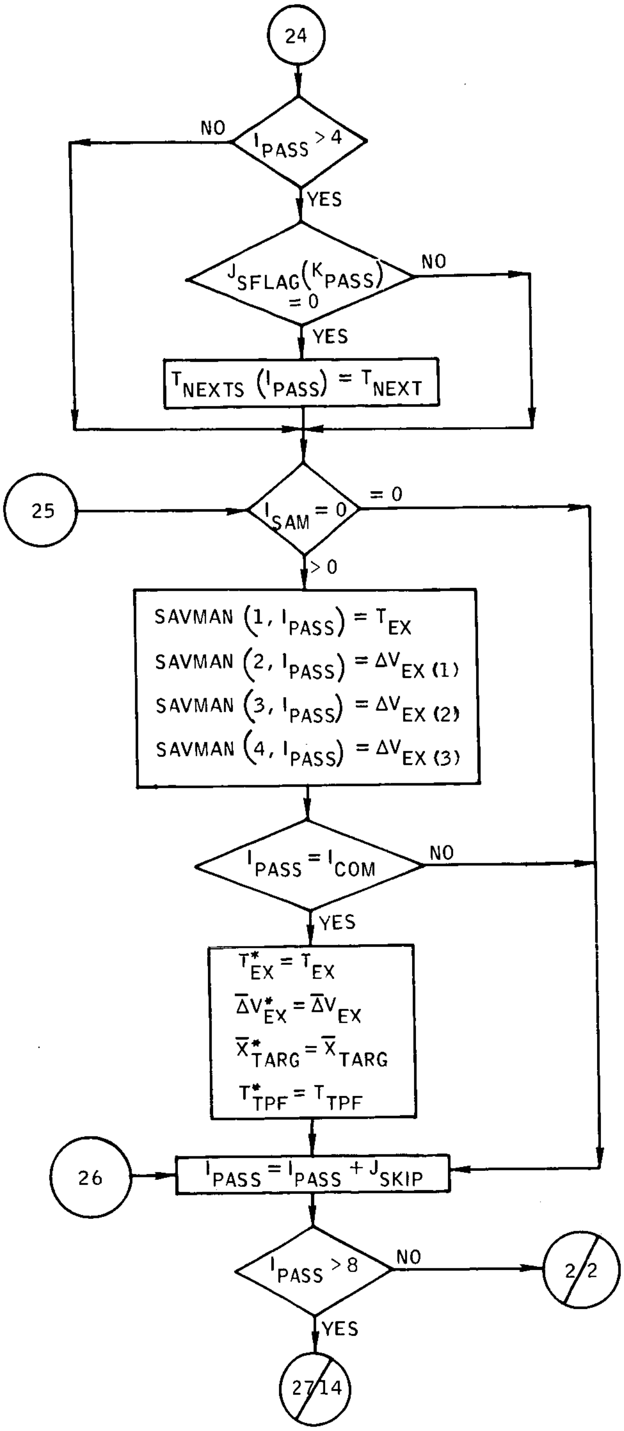
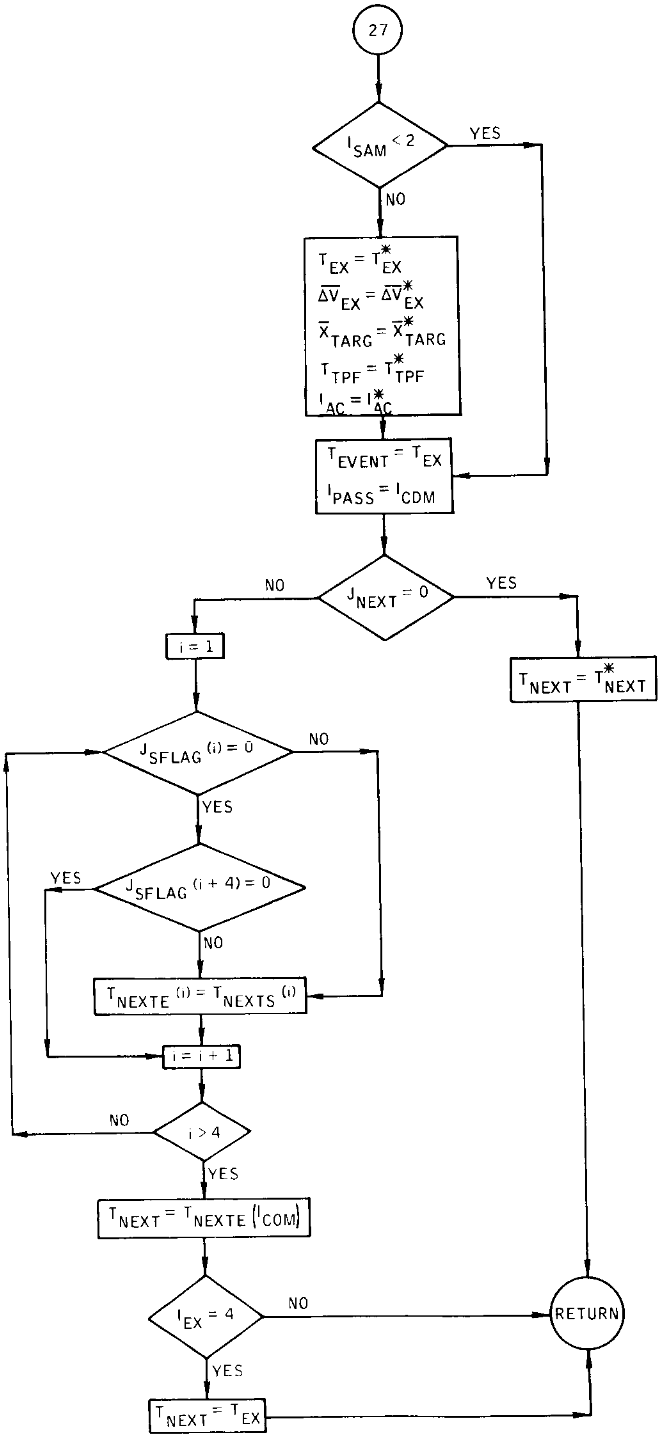

22
I_{SAM}
= 0
> 0
$\overline{U} = \text{UNIT}\left(\overline{R}_J \times \overline{V}_J\right)$
$\text{SAVMAN}\left(5, I_{PASS}\right) = \overline{U} \cdot \overline{R}_I$
I_{WRITE}
> 0
8/7
= 0
23
$I_{PASS} = I_{COM}$
NO
YES
$I^*_{AC} = I_{AC}$
$J_{NEXT} = 0$
YES
NO
$I_{EX} = 4$
NO
YES
$J_{NEXT} = 2$
NO
YES
$T_{NEXT} = T_{EX}$
$I_{PASS} > 4$
YES
24/13
NO
$T_{NEXTS}\left(I_{PASS}\right) = T_{NEXT}$
25/13

Figure B-99.- Continued.


24
$I_{PASS} > 4$
NO
YES
$J_{SFLAG}(K_{PASS}) = 0$
NO
YES
$T_{NEXTS}(I_{PASS}) = T_{NEXT}$
25
$I_{SAM} = 0$
= 0
> 0
SAVMAN $(1, I_{PASS}) = T_{EX}$
SAVMAN $(2, I_{PASS}) = \Delta V_{EX(1)}$
SAVMAN $(3, I_{PASS}) = \Delta V_{EX(2)}$
SAVMAN $(4, I_{PASS}) = \Delta V_{EX(3)}$
$I_{PASS} = I_{COM}$
NO
YES
$T^*_{EX} = T_{EX}$
$\bar{\Delta V}^*_{EX} = \bar{\Delta V}_{EX}$
$\bar{X}^*_{TARG} = \bar{X}_{TARG}$
$T^*_{TPF} = T_{TPF}$
26
$I_{PASS} = I_{PASS} + J_{SKIP}$
$I_{PASS} > 8$
NO
2/2
YES
27/14

Page 13 of 14

Figure B-99.- Continued.



Page 14 of 14

Figure B-99.- Concluded.

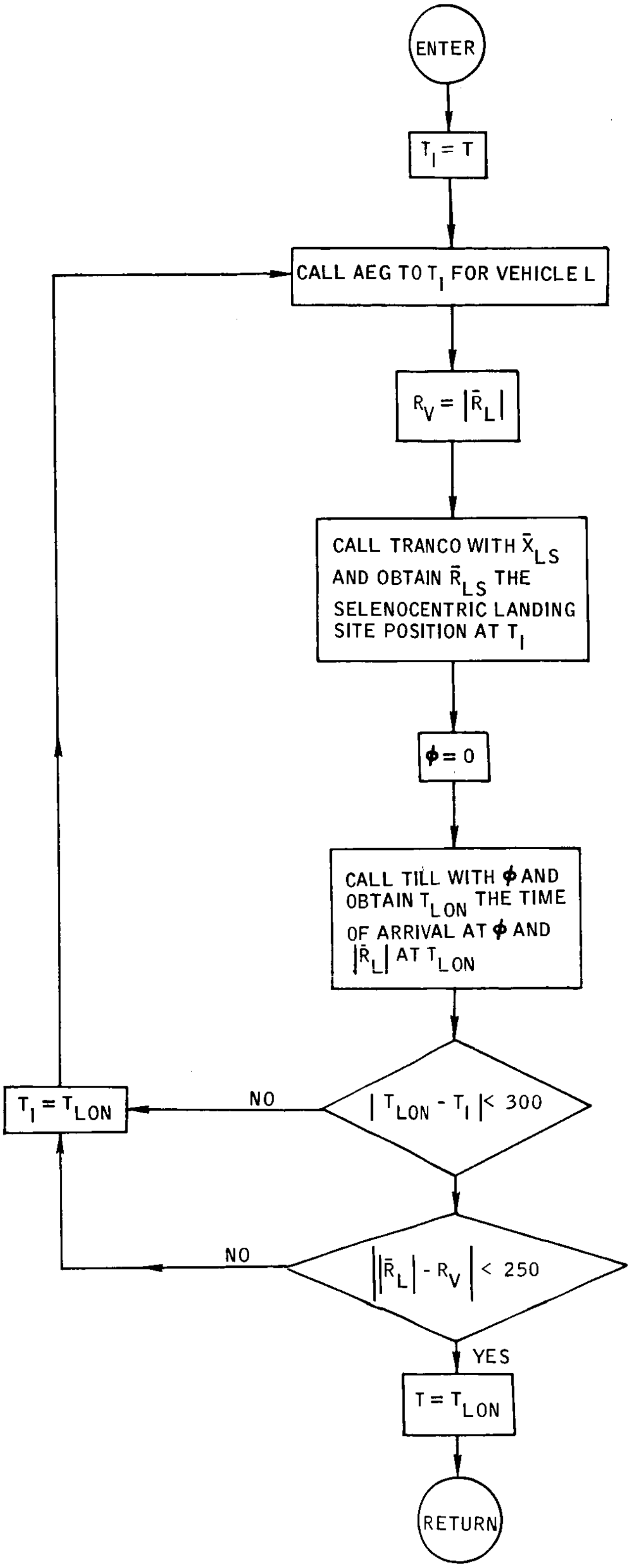
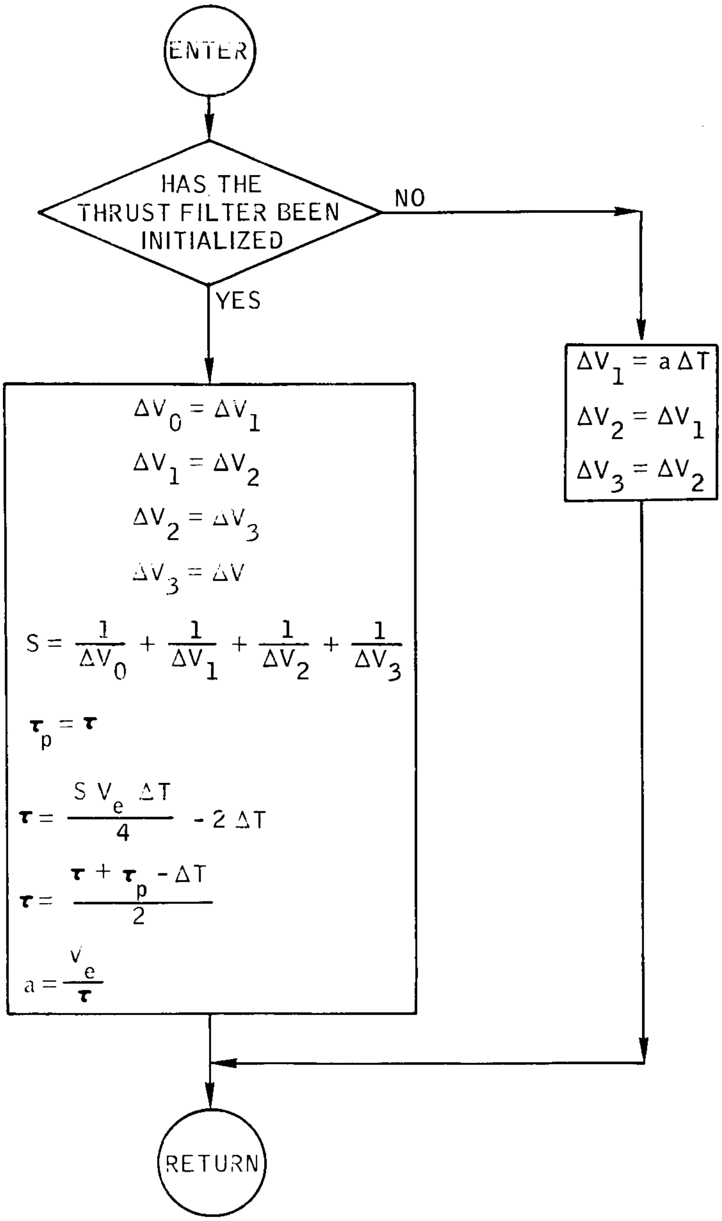

ENTER
$T_I = T$
CALL AEG TO T_I FOR VEHICLE L
$R_V = |\bar{R}_L|$
CALL TRANCO WITH $\bar{X}_{LS}$ AND OBTAIN $\bar{R}_{LS}$ THE SELENOCENTRIC LANDING SITE POSITION AT T_I
$\phi = 0$
CALL TILL WITH ϕ AND OBTAIN T_{LON} THE TIME OF ARRIVAL AT ϕ AND $|\bar{R}_L|$ AT T_{LON}
$|T_{LON} - T_I| < 300$
NO
$T_I = T_{LON}$
$\left||\bar{R}_L| - R_V\right| < 250$
NO
YES
$T = T_{LON}$
RETURN

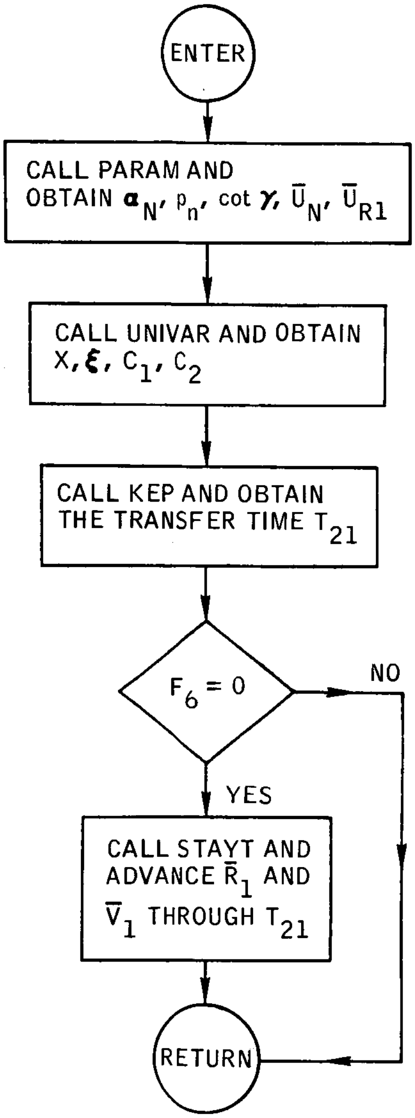
Figure B-100.- Flow chart of subroutine TCONE.

Page 1 of 1


ENTER
HAS THE THRUST FILTER BEEN INITIALIZED
NO
YES
$\Delta V_0 = \Delta V_1$
$\Delta V_1 = \Delta V_2$
$\Delta V_2 = \Delta V_3$
$\Delta V_3 = \Delta V$
$S = \frac{1}{\Delta V_0} + \frac{1}{\Delta V_1} + \frac{1}{\Delta V_2} + \frac{1}{\Delta V_3}$
$\tau_p = \tau$
$\tau = \frac{S V_e \Delta T}{4} - 2 \Delta T$
$\tau = \frac{\tau + \tau_p - \Delta T}{2}$
$a = \frac{V_e}{\tau}$
$\Delta V_1 = a \Delta T$
$\Delta V_2 = \Delta V_1$
$\Delta V_3 = \Delta V_2$
RETURN

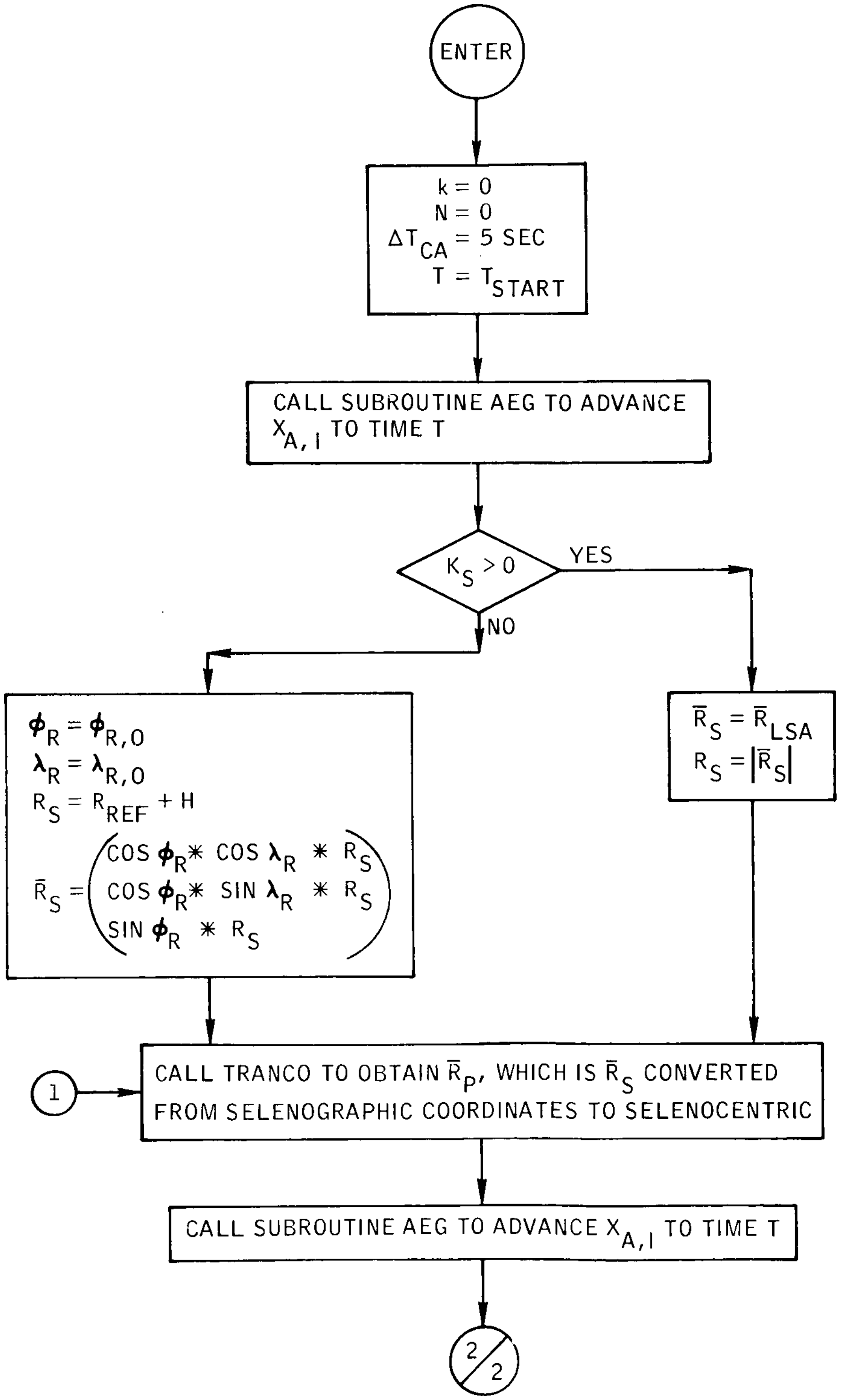
Page 1 of 1

Figure B-101.- Flow chart of subroutine THRFIL.


ENTER
CALL PARAM AND OBTAIN α_N, p_n, cot γ, $\bar{U}_N$, $\bar{U}_{R1}$
CALL UNIVAR AND OBTAIN X, ξ, C_1, C_2
CALL KEP AND OBTAIN THE TRANSFER TIME T_{21}
$F_6 = 0$
NO
YES
CALL STAYT AND ADVANCE $\bar{R}_1$ AND $\bar{V}_1$ THROUGH T_{21}
RETURN

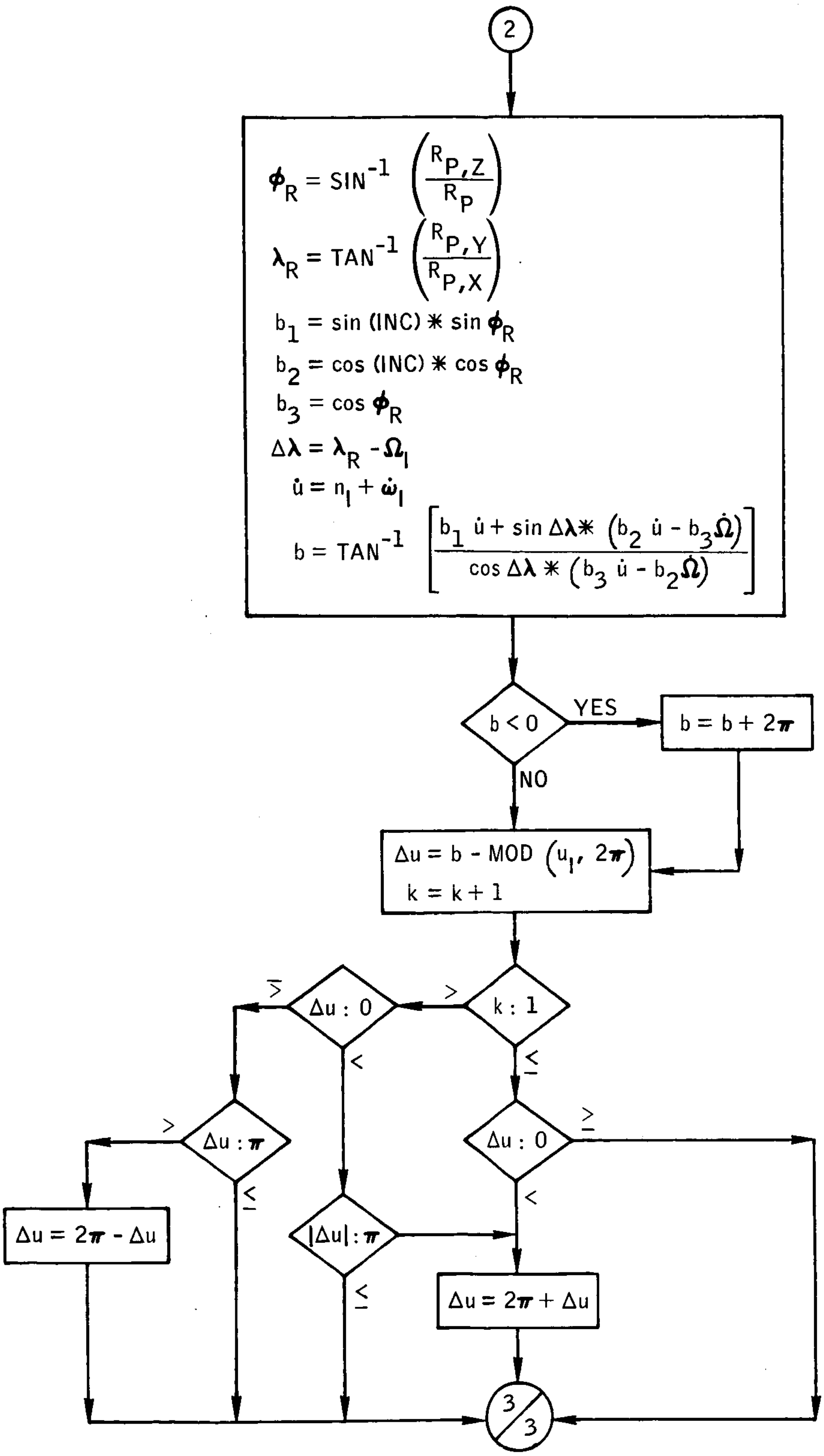
Page 1 of 1

Figure B-1^2.- Flow chart of subroutine TIMEP.


ENTER
$k = 0$
$N = 0$
$\Delta T_{CA} = 5$ SEC
$T = T_{START}$
CALL SUBROUTINE AEG TO ADVANCE $X_{A,I}$ TO TIME T
$K_S > 0$
YES
NO
$\phi_R = \phi_{R,0}$
$\lambda_R = \lambda_{R,0}$
$R_S = R_{REF} + H$
$\bar{R}_S = \begin{pmatrix} \cos \phi_R * \cos \lambda_R * R_S \\ \cos \phi_R * \sin \lambda_R * R_S \\ \sin \phi_R * R_S \end{pmatrix}$
$\bar{R}_S = \bar{R}_{LSA}$
$R_S = |\bar{R}_S|$
1
CALL TRANCO TO OBTAIN $\bar{R}_P$, WHICH IS $\bar{R}_S$ CONVERTED FROM SELENOGRAPHIC COORDINATES TO SELENOCENTRIC
CALL SUBROUTINE AEG TO ADVANCE $X_{A,I}$ TO TIME T
2/2

Page 1 of 3

Figure B-103.- Flow chart of subroutine TPASS.


2
$\phi_R = SIN^{-1}\left(\frac{R_{P,Z}}{R_P}\right)$
$\lambda_R = TAN^{-1}\left(\frac{R_{P,Y}}{R_{P,X}}\right)$
$b_1 = \sin(INC) * \sin\phi_R$
$b_2 = \cos(INC) * \cos\phi_R$
$b_3 = \cos\phi_R$
$\Delta\lambda = \lambda_R - \Omega_I$
$\dot{u} = n_I + \dot{\omega}_I$
$b = TAN^{-1}\left[\frac{b_1\ \dot{u} + \sin\Delta\lambda * (b_2\ \dot{u} - b_3\dot{\Omega})}{\cos\Delta\lambda * (b_3\ \dot{u} - b_2\dot{\Omega})}\right]$
$b < 0$
YES
$b = b + 2\pi$
NO
$\Delta u = b - MOD(u_I, 2\pi)$
$k = k + 1$
$k : 1$
$>$
$\leq$
$\Delta u : 0$
$\geq$
$<$
$\Delta u : \pi$
$>$
$\leq$
$\Delta u = 2\pi - \Delta u$
$|\Delta u| : \pi$
$\leq$
$\Delta u : 0$
$\geq$
$<$
$\Delta u = 2\pi + \Delta u$
3/3

Page 2 of 3

Figure B-103.- Continued.

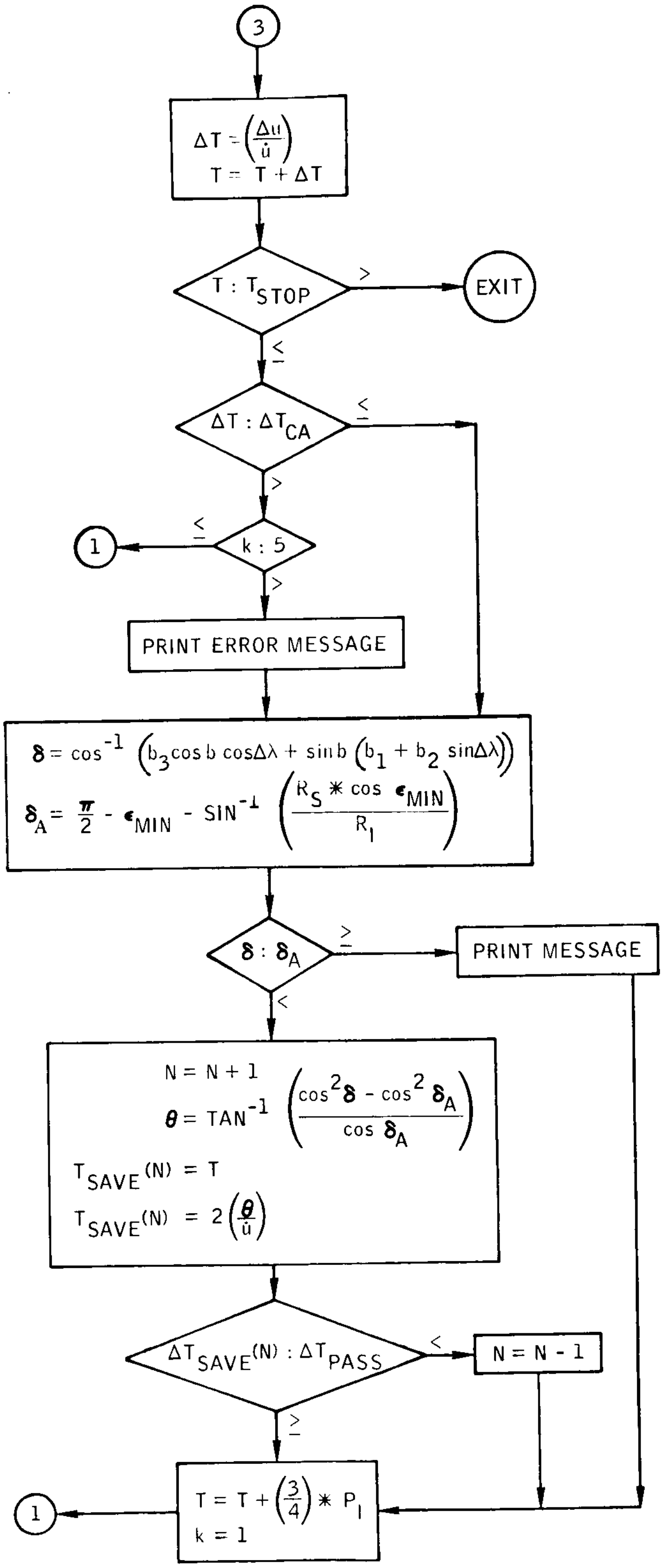

3
$\Delta T = \left(\frac{\Delta u}{\dot{u}}\right)$
$T = T + \Delta T$
$T : T_{STOP}$
$>$
EXIT
$\leq$
$\Delta T : \Delta T_{CA}$
$\leq$
$>$
1
$\leq$
k : 5
$>$
PRINT ERROR MESSAGE
$\delta = \cos^{-1}\left(b_3 \cos b \cos\Delta\lambda + \sin b \left(b_1 + b_2 \sin\Delta\lambda\right)\right)$
$\delta_A = \frac{\pi}{2} - \epsilon_{MIN} - SIN^{-1}\left(\frac{R_S * \cos \epsilon_{MIN}}{R_I}\right)$
$\delta : \delta_A$
$\geq$
PRINT MESSAGE
$<$
$N = N + 1$
$\theta = TAN^{-1}\left(\frac{\cos^2\delta - \cos^2\delta_A}{\cos\delta_A}\right)$
$T_{SAVE}(N) = T$
$T_{SAVE}(N) = 2\left(\frac{\theta}{\dot{u}}\right)$
$\Delta T_{SAVE}(N) : \Delta T_{PASS}$
$<$
$N = N - 1$
$\geq$
$T = T + \left(\frac{3}{4}\right) * P_I$
$k = 1$
1

Figure B-103.- Concluded.

Page 3 of 3

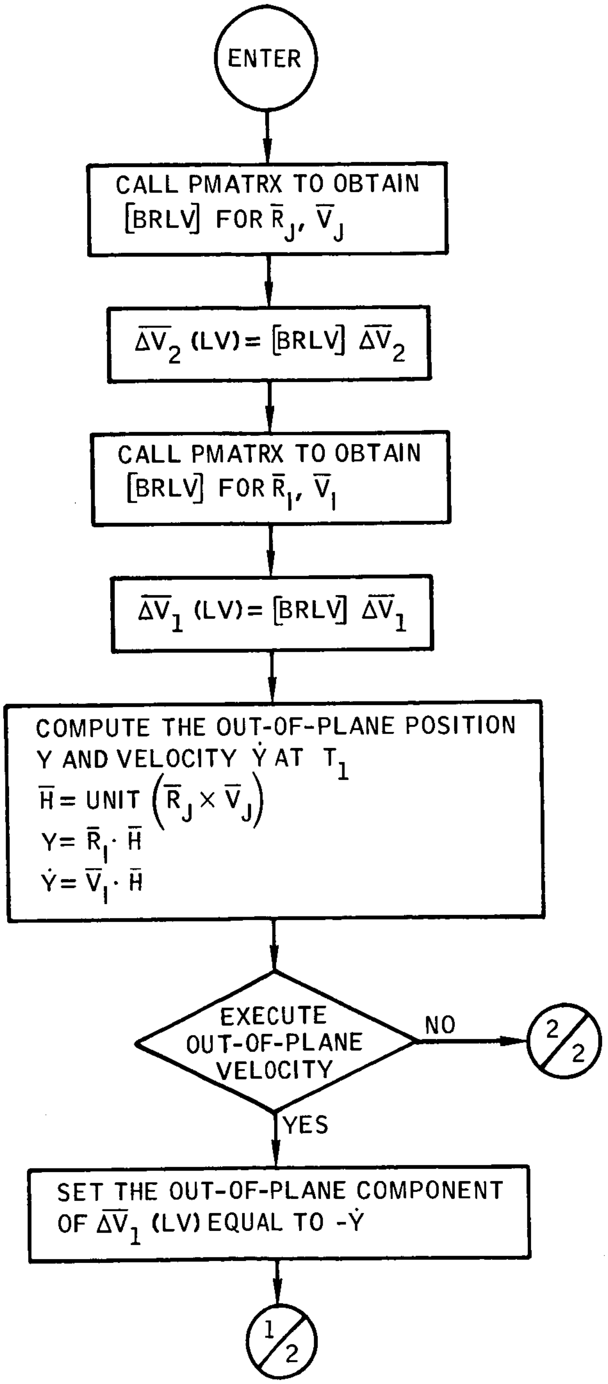
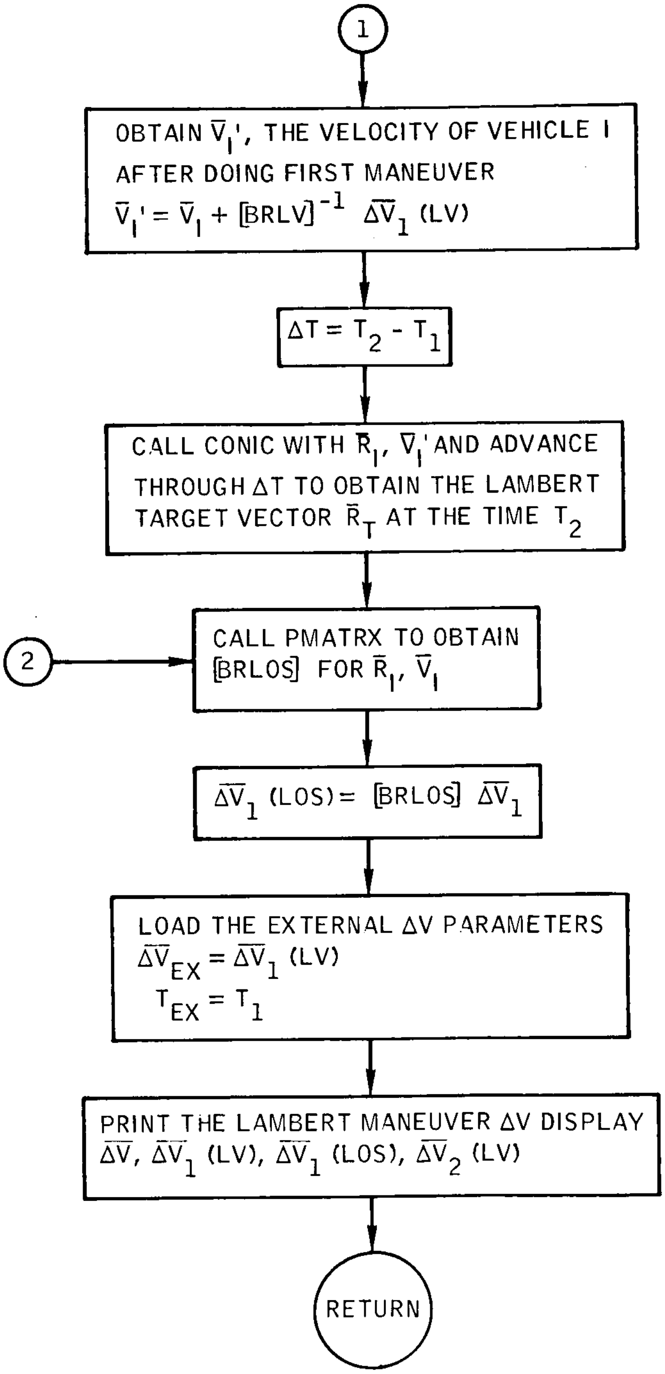

ENTER
CALL PMATRX TO OBTAIN [BRLV] FOR $\bar{R}_J$, $\bar{V}_J$
$\overline{\Delta V}_2$ (LV) = [BRLV] $\overline{\Delta V}_2$
CALL PMATRX TO OBTAIN [BRLV] FOR $\bar{R}_I$, $\bar{V}_I$
$\overline{\Delta V}_1$ (LV) = [BRLV] $\overline{\Delta V}_1$
COMPUTE THE OUT-OF-PLANE POSITION Y AND VELOCITY $\dot{Y}$ AT T_1
$\bar{H}$ = UNIT $\left(\bar{R}_J \times \bar{V}_J\right)$
$Y = \bar{R}_I \cdot \bar{H}$
$\dot{Y} = \bar{V}_I \cdot \bar{H}$
EXECUTE OUT-OF-PLANE VELOCITY
NO
2/2
YES
SET THE OUT-OF-PLANE COMPONENT OF $\overline{\Delta V}_1$ (LV) EQUAL TO $-\dot{Y}$
1/2

Figure B-104.- Flow chart of subroutine TPITPM.

Page 1 of 2


1
OBTAIN $\bar{V}_I'$, THE VELOCITY OF VEHICLE I AFTER DOING FIRST MANEUVER
$\bar{V}_I' = \bar{V}_I + [BRLV]^{-1}\ \Delta\bar{V}_1\ (LV)$
$\Delta T = T_2 - T_1$
CALL CONIC WITH $\bar{R}_I$, $\bar{V}_I'$ AND ADVANCE THROUGH ΔT TO OBTAIN THE LAMBERT TARGET VECTOR $\bar{R}_T$ AT THE TIME T_2
2
CALL PMATRX TO OBTAIN [BRLOS] FOR $\bar{R}_I$, $\bar{V}_I$
$\Delta\bar{V}_1\ (LOS) = [BRLOS]\ \Delta\bar{V}_1$
LOAD THE EXTERNAL ΔV PARAMETERS
$\Delta\bar{V}_{EX} = \Delta\bar{V}_1\ (LV)$
$T_{EX} = T_1$
PRINT THE LAMBERT MANEUVER ΔV DISPLAY
$\Delta\bar{V}$, $\Delta\bar{V}_1\ (LV)$, $\Delta\bar{V}_1\ (LOS)$, $\Delta\bar{V}_2\ (LV)$
RETURN

Page 2 of 2

Figure B-104.- Concluded.

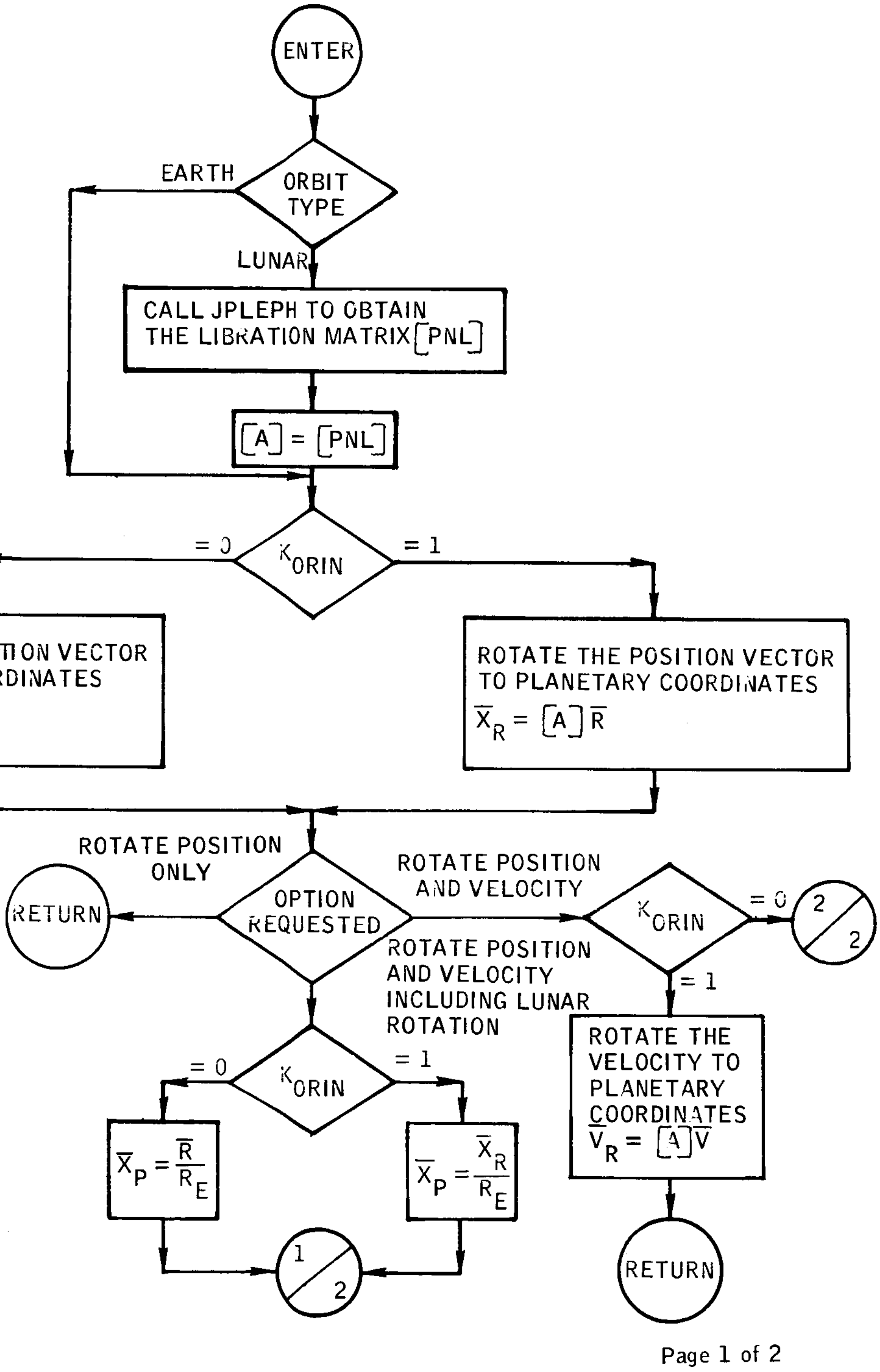
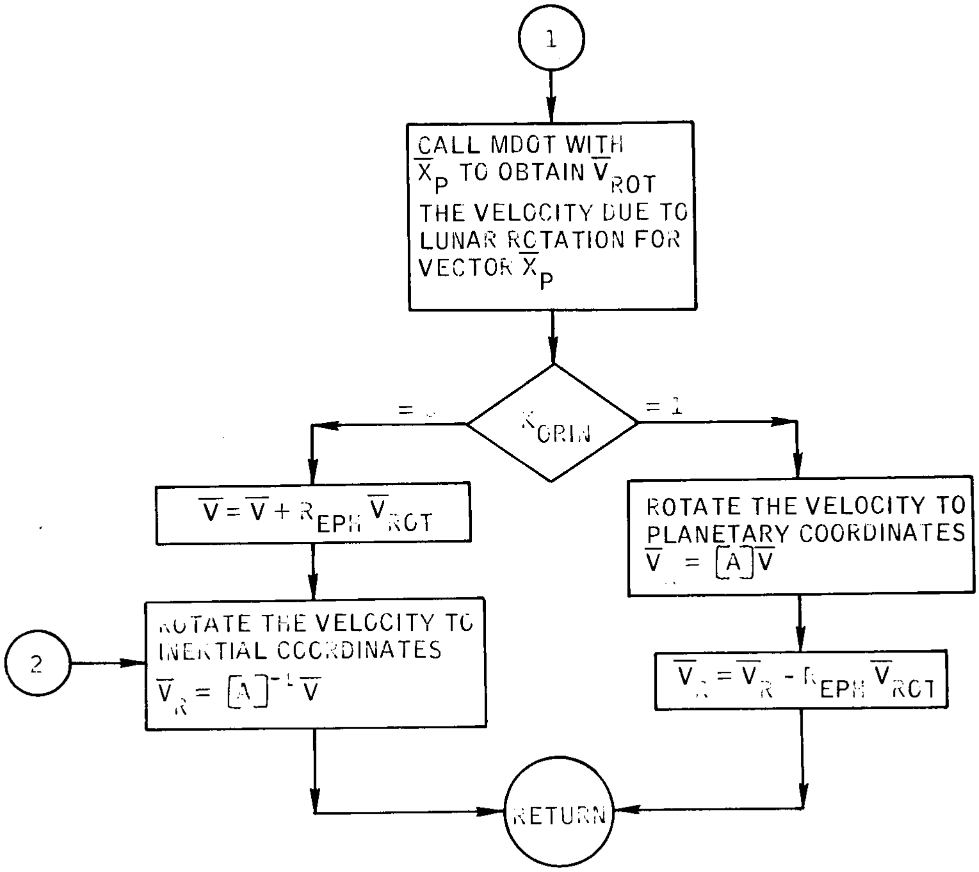
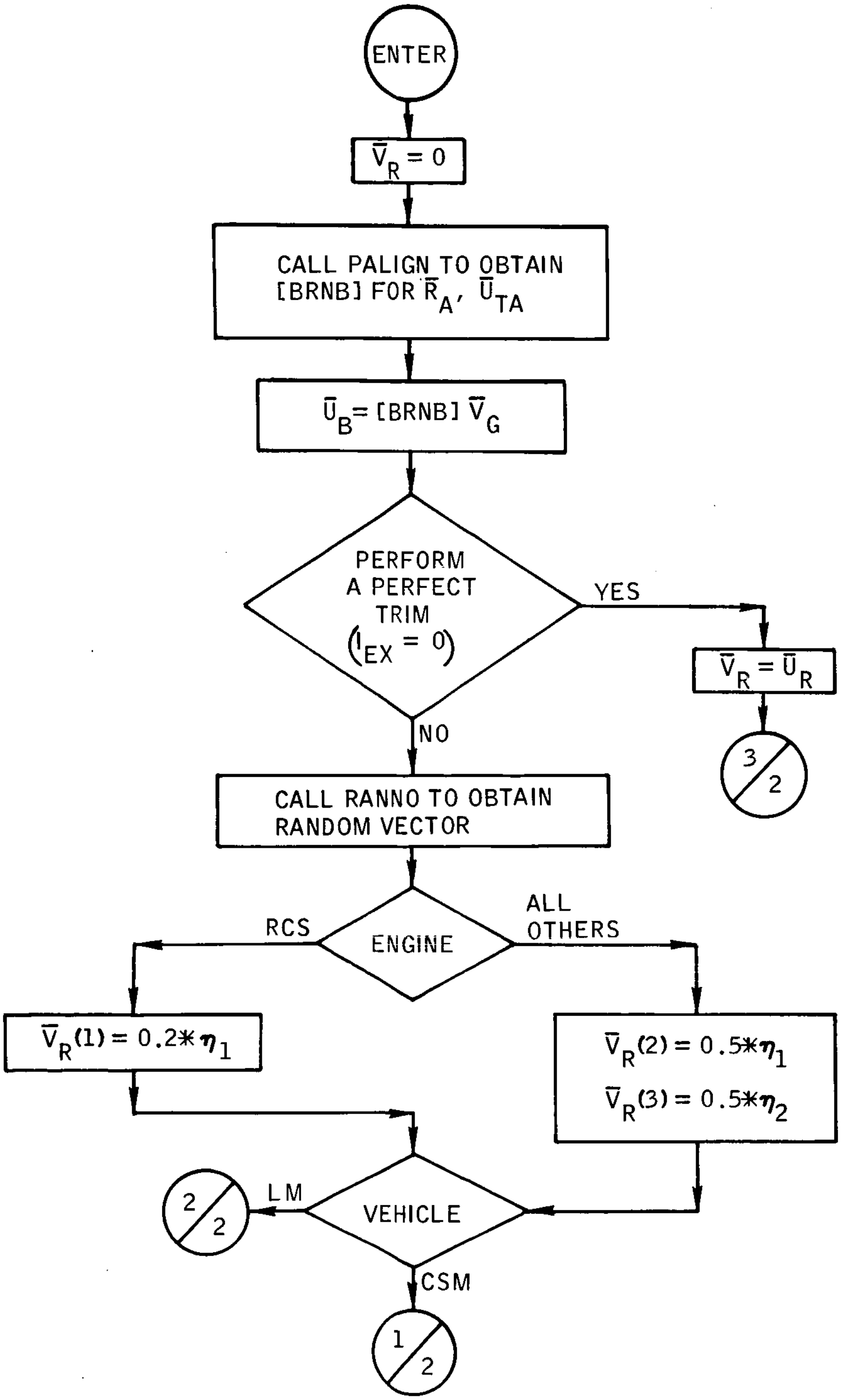

ENTER
ORBIT TYPE
EARTH
LUNAR
CALL JPLEPH TO OBTAIN THE LIBRATION MATRIX [PNL]
[A] = [PNL]
K_{ORIN}
= 0
= 1
ROTATE THE POSITION VECTOR TO INERTIAL COORDINATES $\overline{X}_R = [A]^{-1} \overline{R}$
ROTATE THE POSITION VECTOR TO PLANETARY COORDINATES $\overline{X}_R = [A] \overline{R}$
OPTION REQUESTED
ROTATE POSITION ONLY
RETURN
ROTATE POSITION AND VELOCITY
ROTATE POSITION AND VELOCITY INCLUDING LUNAR ROTATION
K_{ORIN}
= 0
2/2
= 1
ROTATE THE VELOCITY TO PLANETARY COORDINATES $\overline{V}_R = [A] \overline{V}$
RETURN
K_{ORIN}
= 0
= 1
$\overline{X}_P = \frac{\overline{R}}{R_E}$
$\overline{X}_P = \frac{\overline{X}_R}{R_E}$
1/2
Page 1 of 2

Figure B-105.- Flow chart of subroutine TRANCO.


1
CALL MDOT WITH $\overline{X}_P$ TO OBTAIN $\overline{V}_{ROT}$ THE VELOCITY DUE TO LUNAR ROTATION FOR VECTOR $\overline{X}_P$
K_{ORIN}
= 0
= 1
$\overline{V} = \overline{V} + R_{EPH}\,\overline{V}_{ROT}$
ROTATE THE VELOCITY TO PLANETARY COORDINATES
$\overline{V} = [A]\overline{V}$
2
ROTATE THE VELOCITY TO INERTIAL COORDINATES
$\overline{V}_R = [A]^{-1}\,\overline{V}$
$\overline{V}_R = \overline{V}_R - R_{EPH}\,\overline{V}_{ROT}$
RETURN

Page 2 of 2

Figure B-105.- Concluded.


ENTER
$\bar{V}_R = 0$
CALL PALIGN TO OBTAIN [BRNB] FOR $\bar{R}_A$, $\bar{U}_{TA}$
$\bar{U}_B$= [BRNB] $\bar{V}_G$
PERFORM A PERFECT TRIM (I_{EX} = 0)
YES
$\bar{V}_R = \bar{U}_R$
3/2
NO
CALL RANNO TO OBTAIN RANDOM VECTOR
ENGINE
RCS
ALL OTHERS
$\bar{V}_R(1) = 0.2*\eta_1$
$\bar{V}_R(2) = 0.5*\eta_1$
$\bar{V}_R(3) = 0.5*\eta_2$
VEHICLE
LM
2/2
CSM
1/2

Page 1 of 3

Figure B-106.- Flow chart of subroutine TRIM.

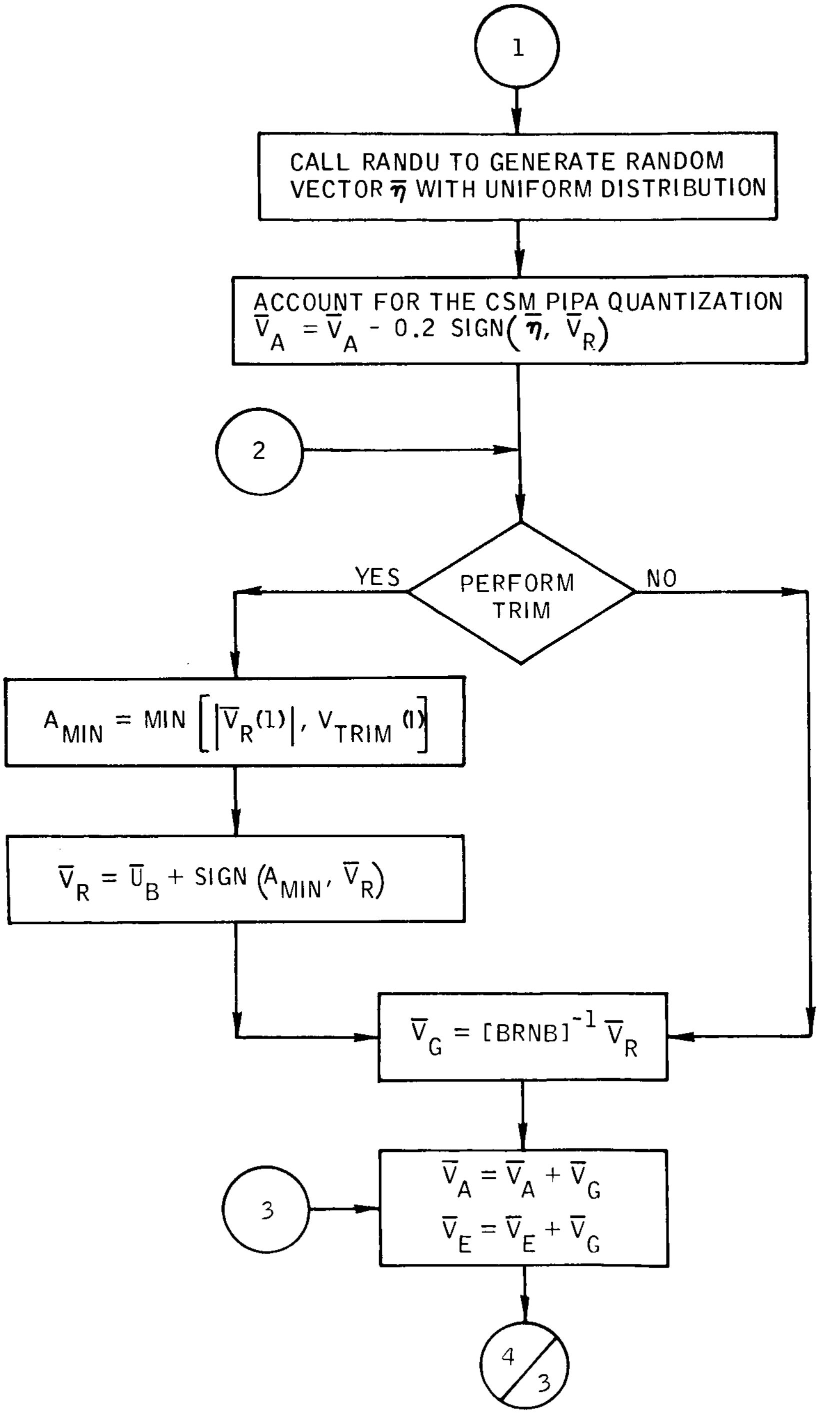
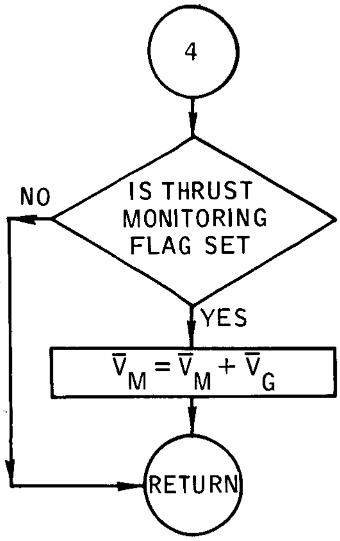

1
CALL RANDU TO GENERATE RANDOM VECTOR $\overline{\eta}$ WITH UNIFORM DISTRIBUTION
ACCOUNT FOR THE CSM PIPA QUANTIZATION $\overline{V}_A = \overline{V}_A - 0.2\ \mathrm{SIGN}(\overline{\eta}, \overline{V}_R)$
2
PERFORM TRIM
YES
NO
$A_{MIN} = \mathrm{MIN}\left[\left|\overline{V}_R(1)\right|, V_{TRIM}(1)\right]$
$\overline{V}_R = \overline{U}_B + \mathrm{SIGN}(A_{MIN}, \overline{V}_R)$
$\overline{V}_G = [BRNB]^{-1}\, \overline{V}_R$
3
$\overline{V}_A = \overline{V}_A + \overline{V}_G$
$\overline{V}_E = \overline{V}_E + \overline{V}_G$
4/3

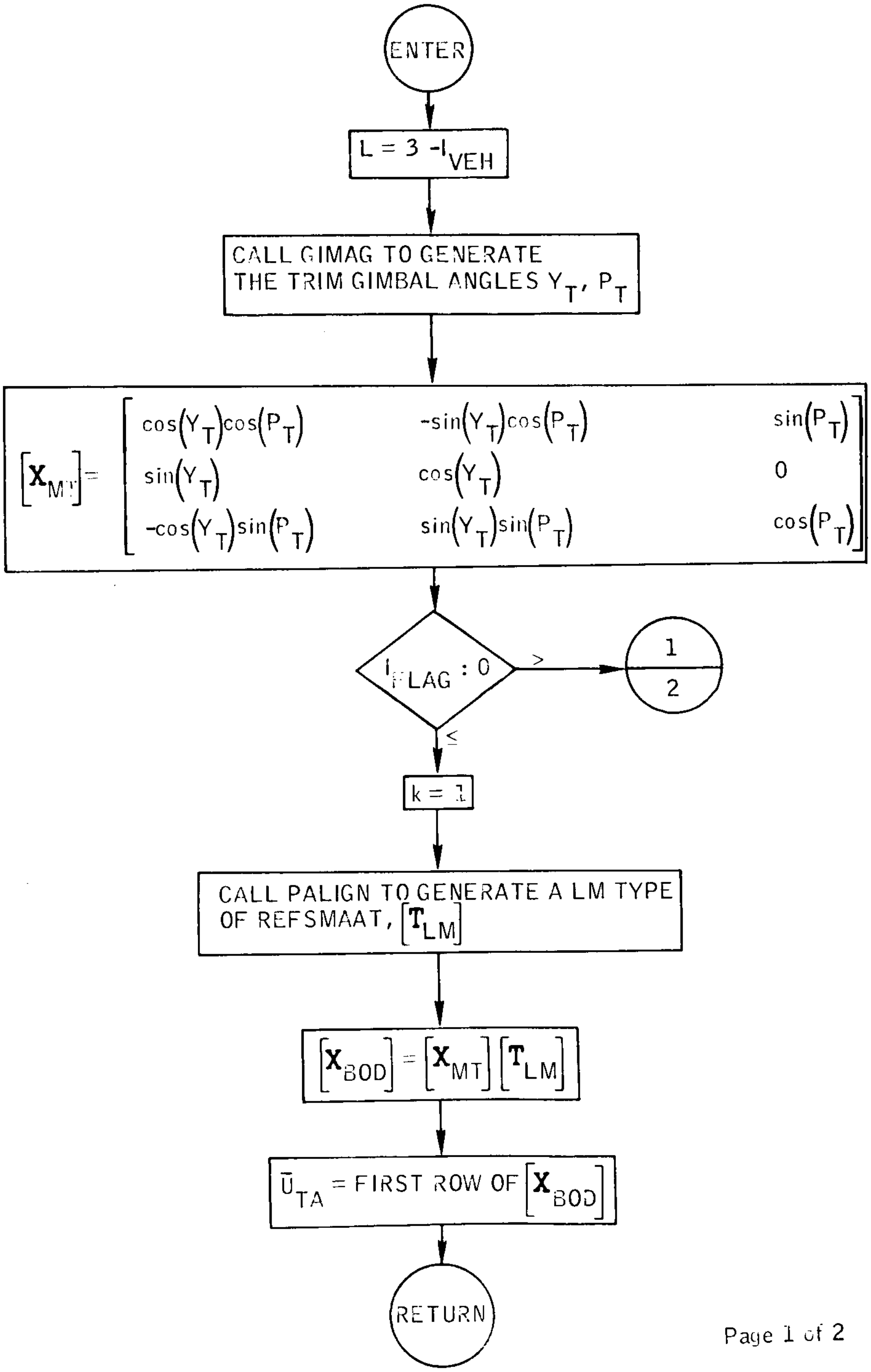
Figure B-106.- Continued.

Page 2 of 3


4
IS THRUST MONITORING FLAG SET
NO
YES
$\overline{V}_M = \overline{V}_M + \overline{V}_G$
RETURN

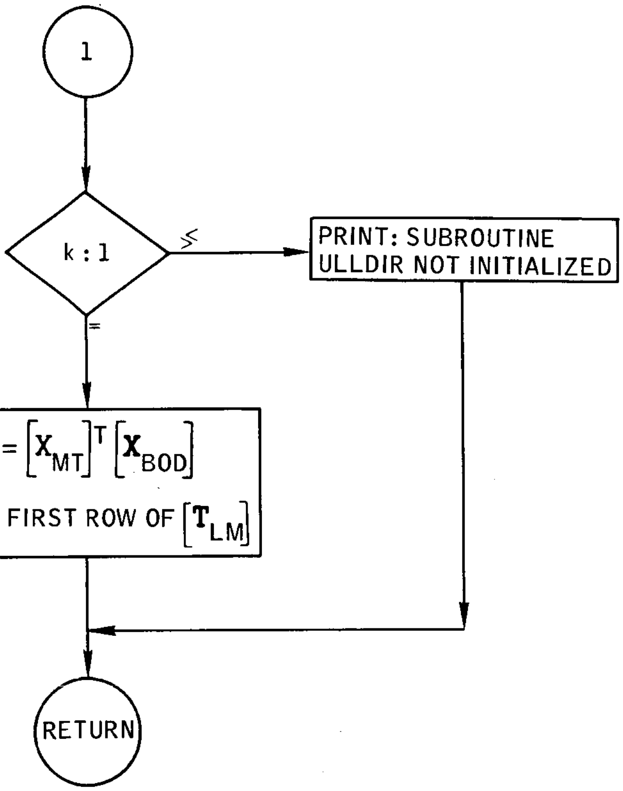
Page 3 of 3

Figure B-106.- Concluded.


ENTER
$L = 3 - I_{VEH}$
CALL GIMAG TO GENERATE THE TRIM GIMBAL ANGLES Y_T, P_T
$[X_{MT}] = \begin{bmatrix} \cos(Y_T)\cos(P_T) & -\sin(Y_T)\cos(P_T) & \sin(P_T) \\ \sin(Y_T) & \cos(Y_T) & 0 \\ -\cos(Y_T)\sin(P_T) & \sin(Y_T)\sin(P_T) & \cos(P_T) \end{bmatrix}$
$I_{FLAG} : 0$
>
1
2
≤
k = 1
CALL PALIGN TO GENERATE A LM TYPE OF REFSMAAT, $[T_{LM}]$
$[X_{BOD}] = [X_{MT}] [T_{LM}]$
$\bar{U}_{TA}$ = FIRST ROW OF $[X_{BOD}]$
RETURN

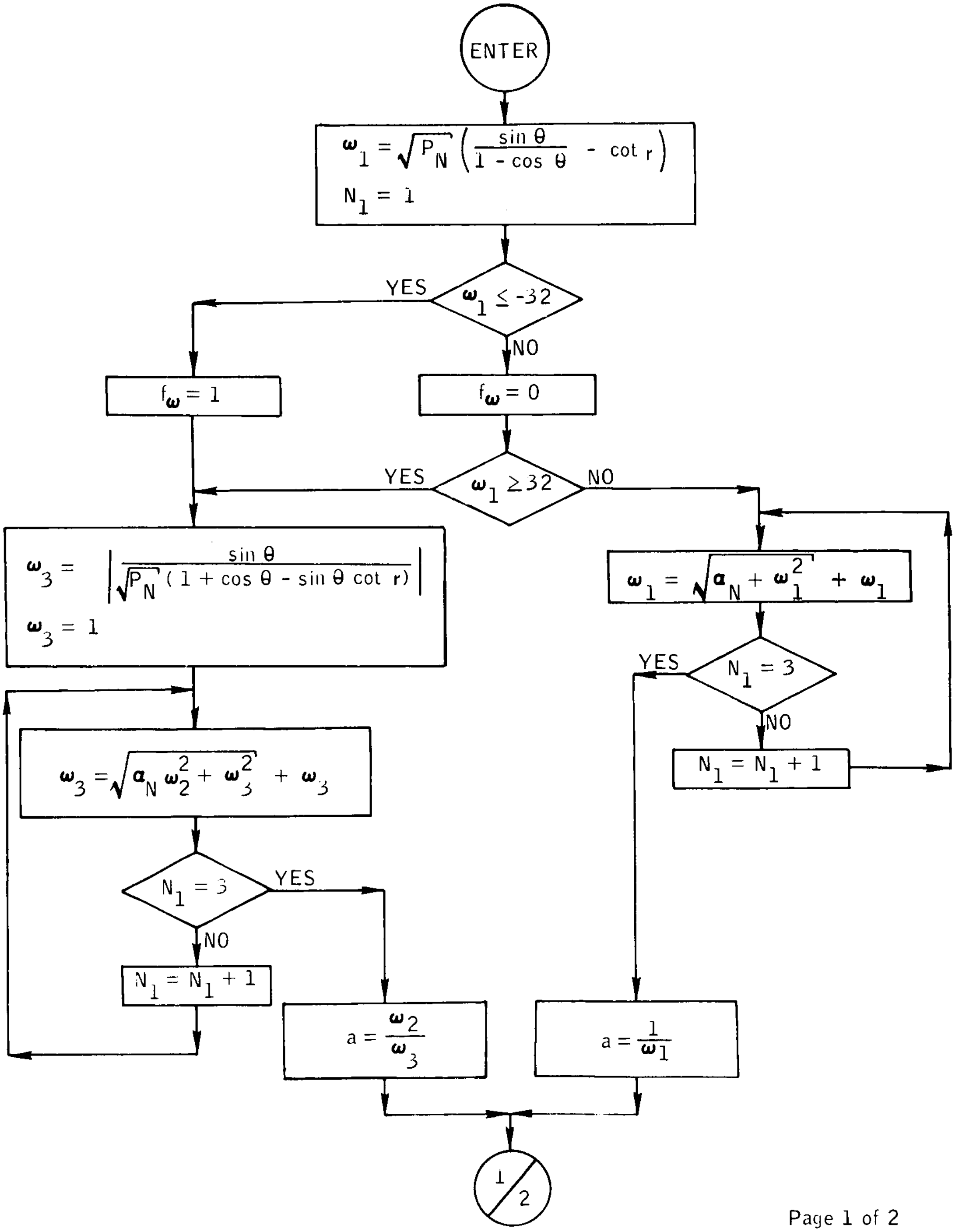
Page 1 of 2

Figure B-107.- Flow chart of subroutine ULLDIR.


1
k : 1
≤
PRINT: SUBROUTINE ULLDIR NOT INITIALIZED
=
$\left[\mathbf{T}_{LM}\right] = \left[\mathbf{X}_{MT}\right]^T \left[\mathbf{X}_{BOD}\right]$
$\bar{U}_{TA}$ = FIRST ROW OF $\left[\mathbf{T}_{LM}\right]$
RETURN

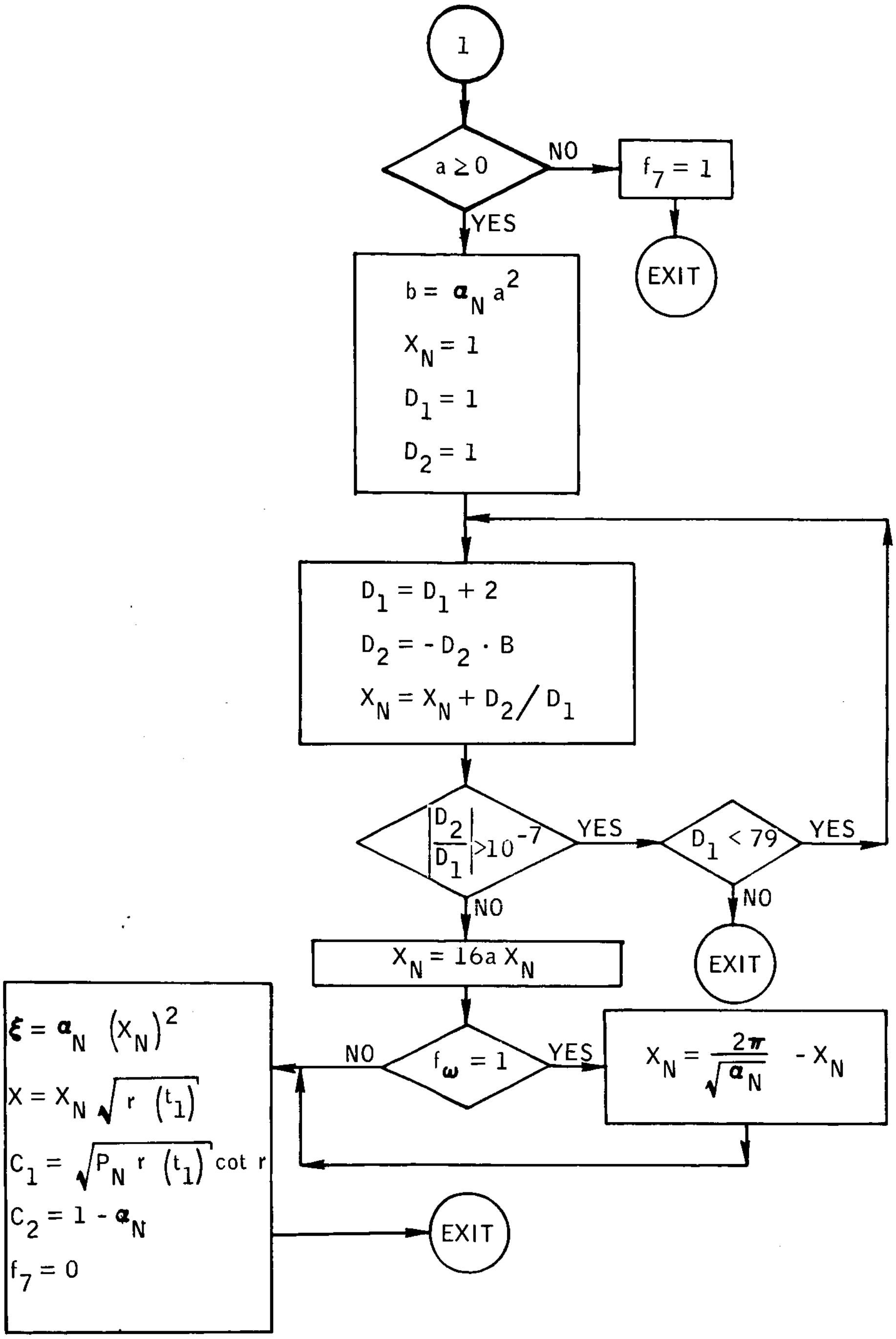
Page 2 of 2

Figure B-107.- Concluded.


ENTER
$\omega_1 = \sqrt{P_N}\left(\frac{\sin\theta}{1 - \cos\theta} - \cot r\right)$
$N_1 = 1$
$\omega_1 \leq -32$
YES
NO
$f_\omega = 1$
$f_\omega = 0$
$\omega_1 \geq 32$
YES
NO
$\omega_3 = \left|\frac{\sin\theta}{\sqrt{P_N}(1 + \cos\theta - \sin\theta \cot r)}\right|$
$\omega_3 = 1$
$\omega_1 = \sqrt{\alpha_N + \omega_1^2} + \omega_1$
$N_1 = 3$
YES
NO
$N_1 = N_1 + 1$
$\omega_3 = \sqrt{\alpha_N \omega_2^2 + \omega_3^2} + \omega_3$
$N_1 = 3$
YES
NO
$N_1 = N_1 + 1$
$a = \frac{\omega_2}{\omega_3}$
$a = \frac{1}{\omega_1}$
1
2

Page 1 of 2

Figure B-108.- Flow chart of subroutine UNIVAR.



Page 2 of 2

Figure B-108.- Concluded.

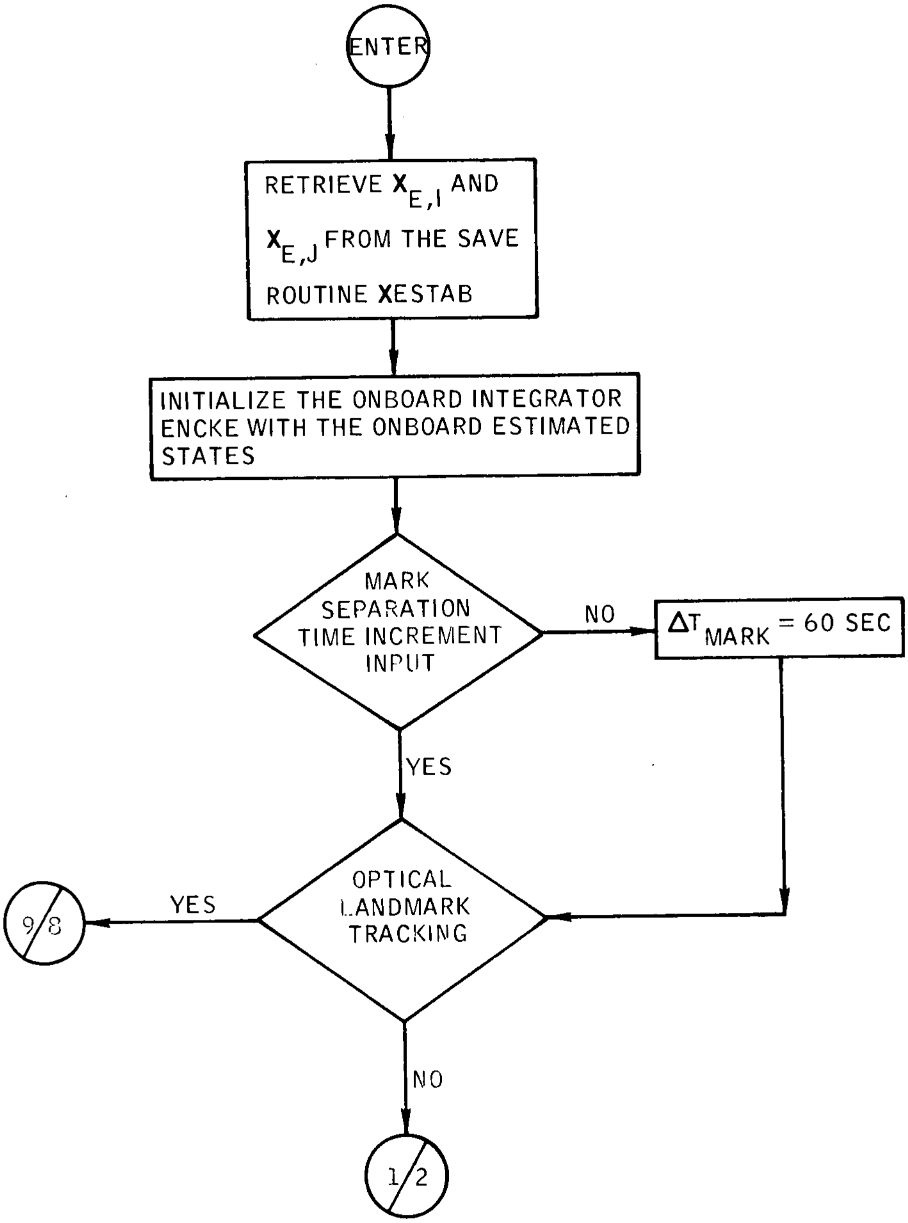
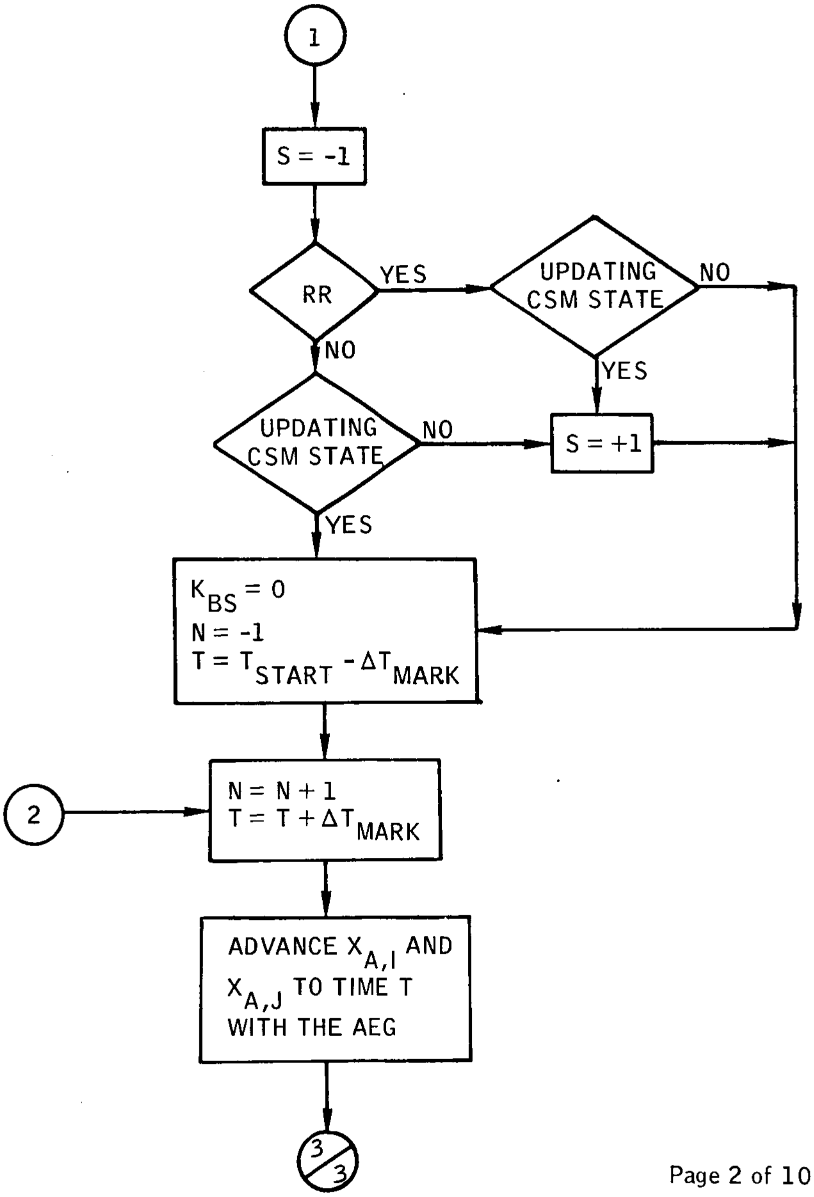

ENTER
RETRIEVE $X_{E,I}$ AND $X_{E,J}$ FROM THE SAVE ROUTINE XESTAB
INITIALIZE THE ONBOARD INTEGRATOR ENCKE WITH THE ONBOARD ESTIMATED STATES
MARK SEPARATION TIME INCREMENT INPUT
NO
ΔT_{MARK} = 60 SEC
YES
OPTICAL LANDMARK TRACKING
YES
9/8
NO
1/2

Figure B-109.- Flow chart of subroutine UPDATV.

Page 1 of 10


1
S = -1
RR
YES
NO
UPDATING CSM STATE
NO
YES
UPDATING CSM STATE
NO
YES
S = +1
$K_{BS} = 0$
N = -1
$T = T_{START} - \Delta T_{MARK}$
2
N = N + 1
$T = T + \Delta T_{MARK}$
ADVANCE $X_{A,I}$ AND $X_{A,J}$ TO TIME T WITH THE AEG
3
3

Page 2 of 10

Figure B-109.- Continued.

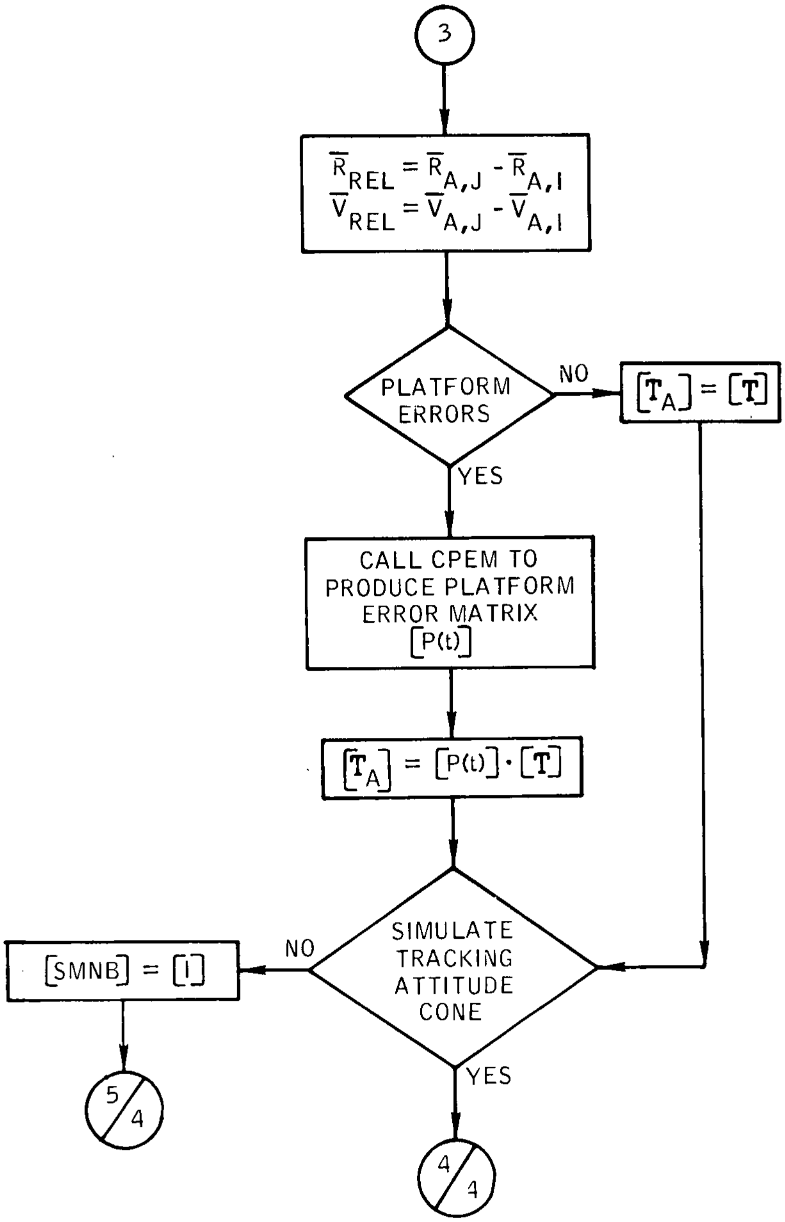

3
$\overline{R}_{REL} = \overline{R}_{A,J} - \overline{R}_{A,I}$
$\overline{V}_{REL} = \overline{V}_{A,J} - \overline{V}_{A,I}$
PLATFORM ERRORS
NO
$[T_A] = [T]$
YES
CALL CPEM TO PRODUCE PLATFORM ERROR MATRIX $[P(t)]$
$[T_A] = [P(t)] \cdot [T]$
SIMULATE TRACKING ATTITUDE CONE
NO
$[SMNB] = [I]$
YES
5/4
4/4

Figure B-109.- Continued.

Page 3 of 10

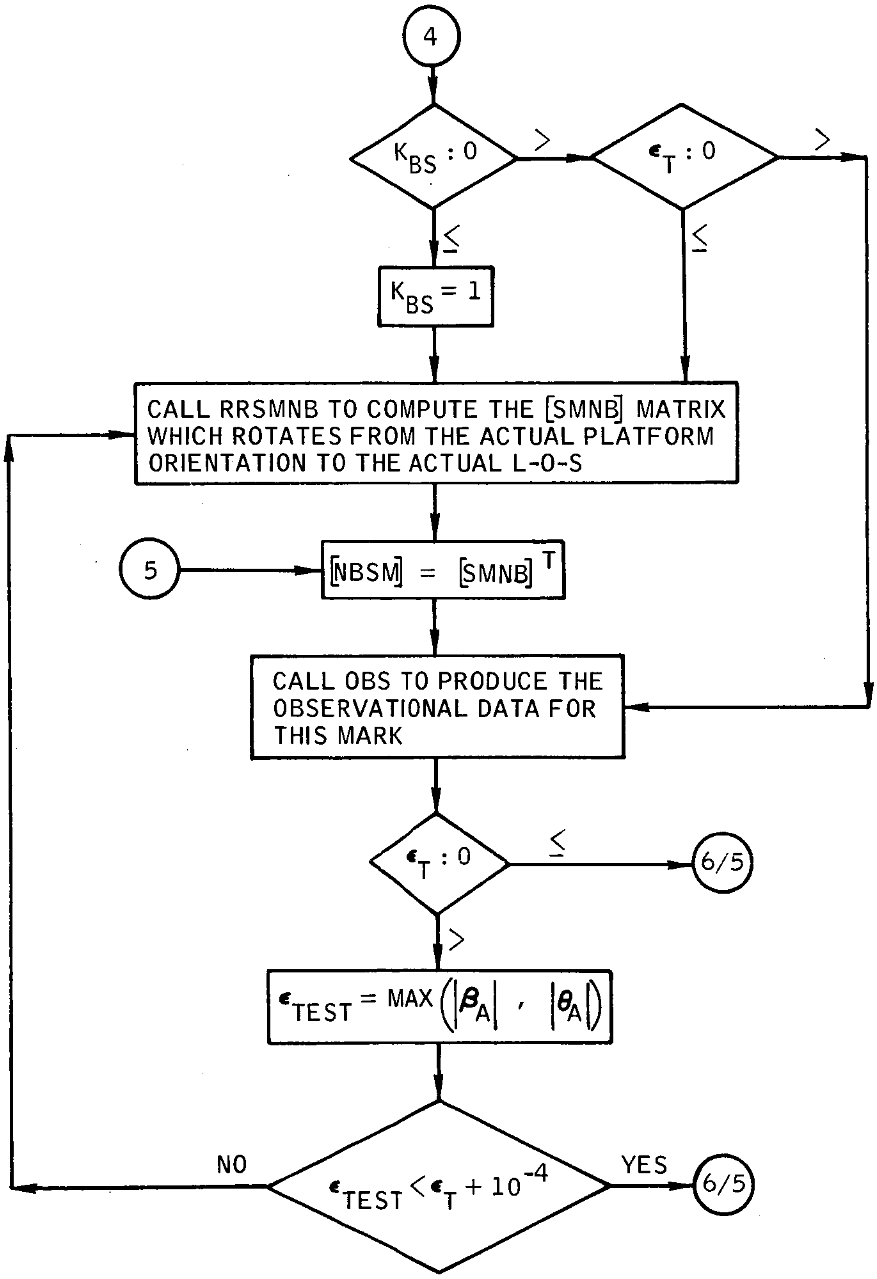
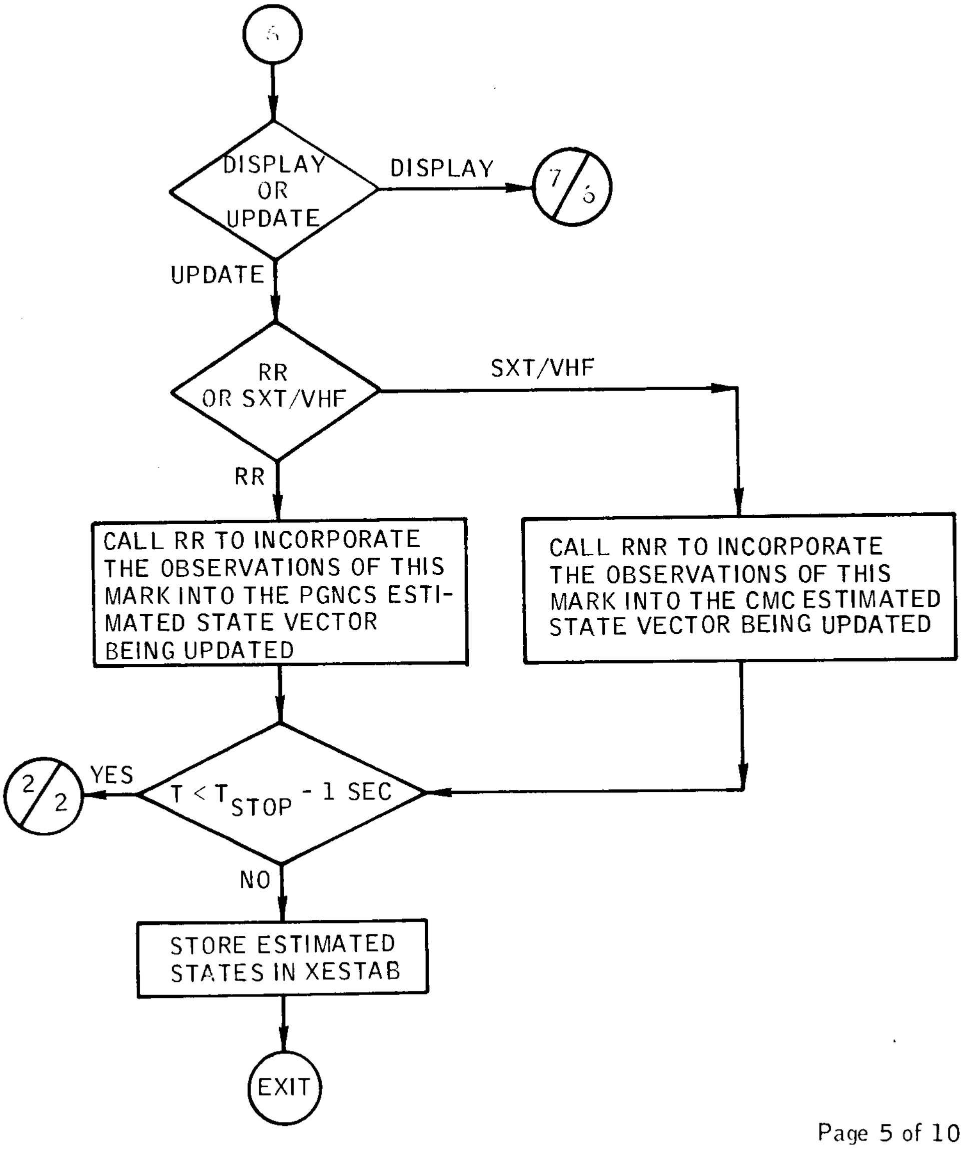

4
$K_{BS} : 0$
>
$\epsilon_T : 0$
>
≤
≤
$K_{BS} = 1$
CALL RRSMNB TO COMPUTE THE [SMNB] MATRIX WHICH ROTATES FROM THE ACTUAL PLATFORM ORIENTATION TO THE ACTUAL L-0-S
5
$[NBSM] = [SMNB]^T$
CALL OBS TO PRODUCE THE OBSERVATIONAL DATA FOR THIS MARK
$\epsilon_T : 0$
≤
6/5
>
$\epsilon_{TEST} = MAX\left(|\beta_A|, |\theta_A|\right)$
NO
$\epsilon_{TEST} < \epsilon_T + 10^{-4}$
YES
6/5

Figure B-109.- Continued.


6
DISPLAY OR UPDATE
DISPLAY
7/6
UPDATE
RR OR SXT/VHF
SXT/VHF
RR
CALL RR TO INCORPORATE THE OBSERVATIONS OF THIS MARK INTO THE PGNCS ESTIMATED STATE VECTOR BEING UPDATED
CALL RNR TO INCORPORATE THE OBSERVATIONS OF THIS MARK INTO THE CMC ESTIMATED STATE VECTOR BEING UPDATED
T < T STOP - 1 SEC
YES
2/2
NO
STORE ESTIMATED STATES IN XESTAB
EXIT

Page 5 of 10

Figure B-109.- Continued.

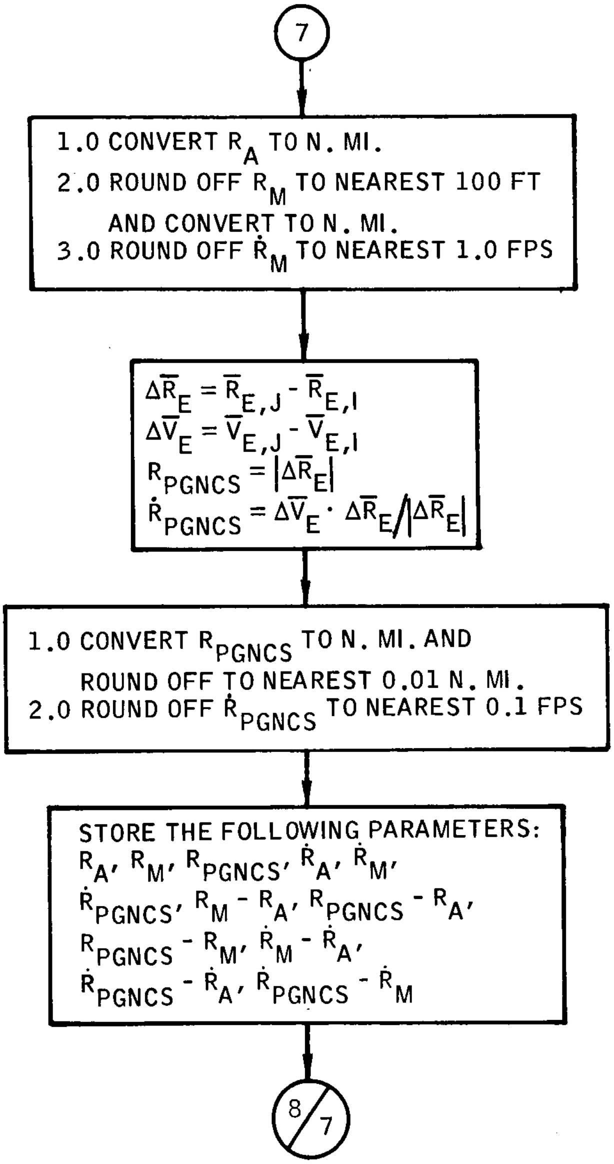

7
1.0 CONVERT R_A TO N. MI.
2.0 ROUND OFF R_M TO NEAREST 100 FT AND CONVERT TO N. MI.
3.0 ROUND OFF $\dot{R}_M$ TO NEAREST 1.0 FPS
$\Delta\overline{R}_E = \overline{R}_{E,J} - \overline{R}_{E,I}$
$\Delta\overline{V}_E = \overline{V}_{E,J} - \overline{V}_{E,I}$
$R_{PGNCS} = |\Delta\overline{R}_E|$
$\dot{R}_{PGNCS} = \Delta\overline{V}_E \cdot \Delta\overline{R}_E / |\Delta\overline{R}_E|$
1.0 CONVERT R_{PGNCS} TO N. MI. AND ROUND OFF TO NEAREST 0.01 N. MI.
2.0 ROUND OFF $\dot{R}_{PGNCS}$ TO NEAREST 0.1 FPS
STORE THE FOLLOWING PARAMETERS:
R_A, R_M, R_{PGNCS}, $\dot{R}_A$, $\dot{R}_M$, $\dot{R}_{PGNCS}$, $R_M - R_A$, $R_{PGNCS} - R_A$, $R_{PGNCS} - R_M$, $\dot{R}_M - \dot{R}_A$, $\dot{R}_{PGNCS} - \dot{R}_A$, $\dot{R}_{PGNCS} - \dot{R}_M$
8/7

Figure B-109.- Continued.

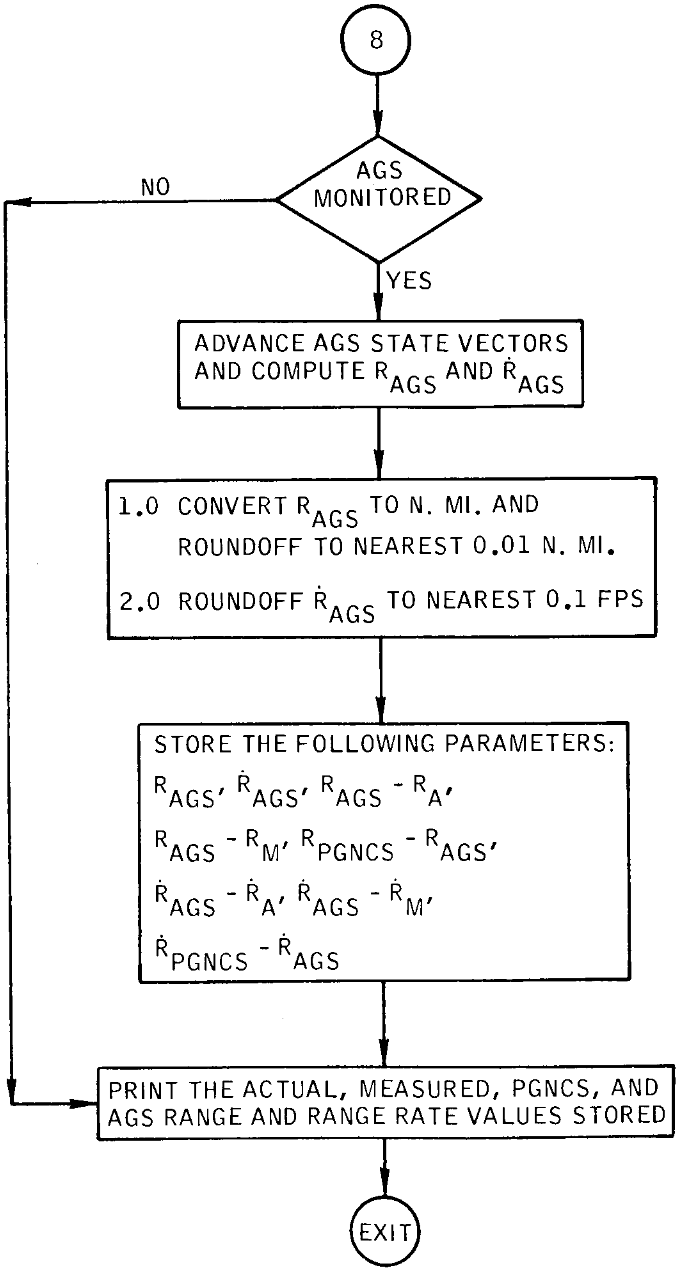
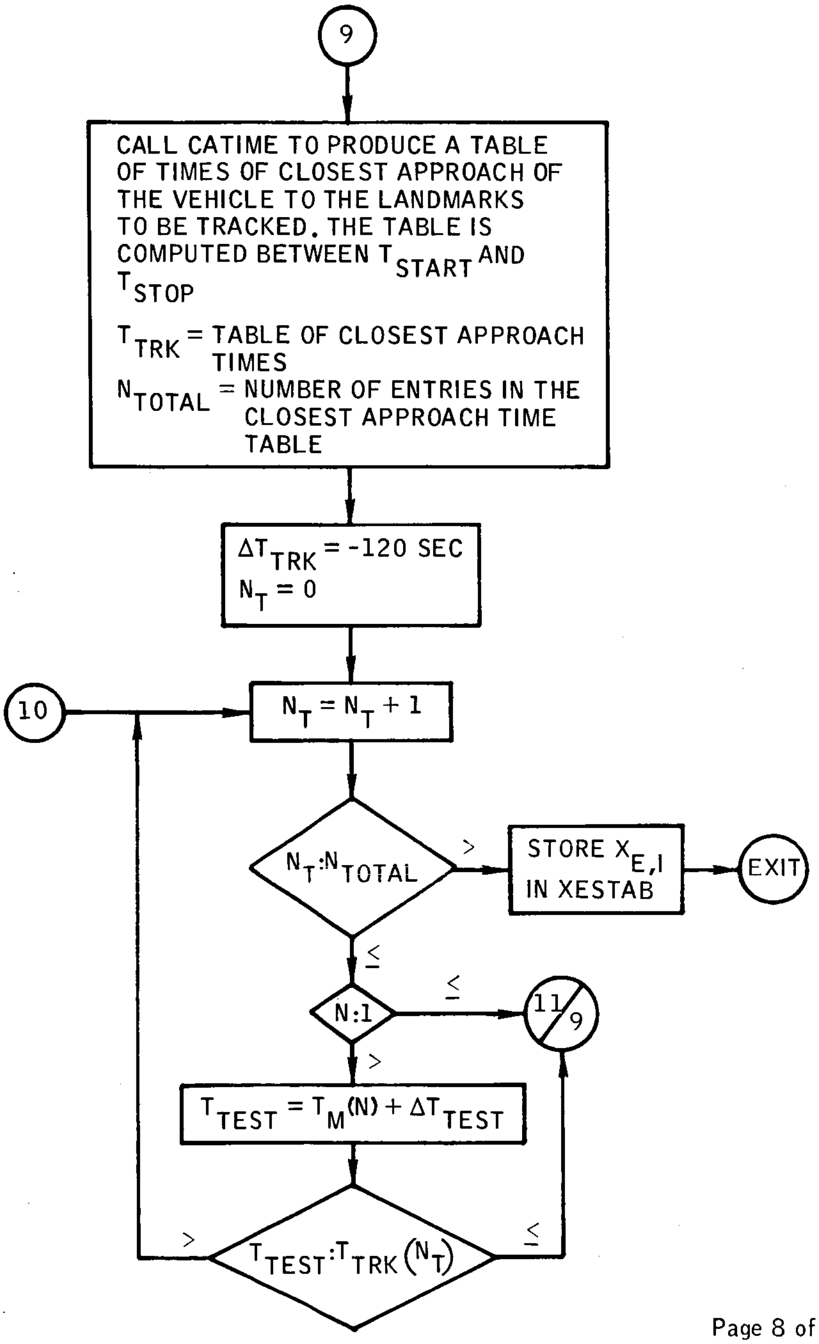
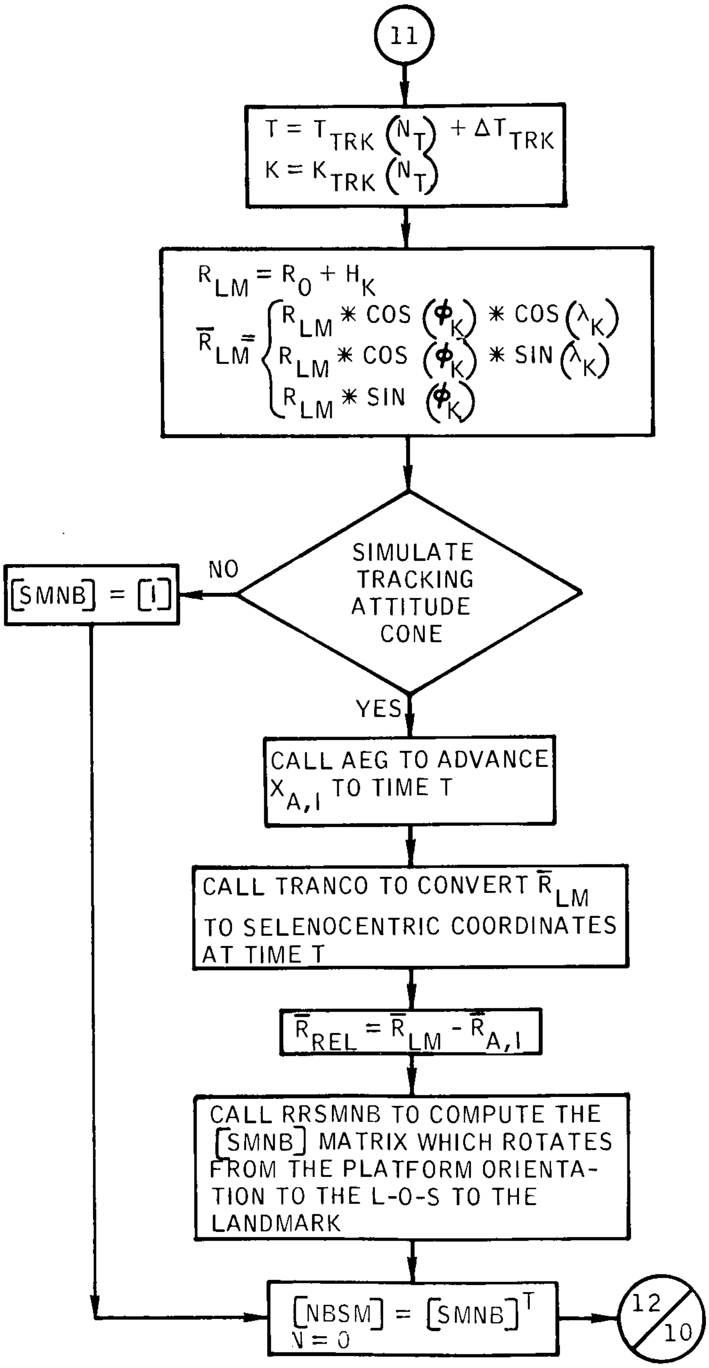

8
AGS MONITORED
NO
YES
ADVANCE AGS STATE VECTORS AND COMPUTE R_{AGS} AND $\dot{R}_{AGS}$
1.0 CONVERT R_{AGS} TO N. MI. AND ROUNDOFF TO NEAREST 0.01 N. MI.
2.0 ROUNDOFF $\dot{R}_{AGS}$ TO NEAREST 0.1 FPS
STORE THE FOLLOWING PARAMETERS:
R_{AGS}, $\dot{R}_{AGS}$, $R_{AGS} - R_{A}$,
$R_{AGS} - R_{M}$, $R_{PGNCS} - R_{AGS}$,
$\dot{R}_{AGS} - \dot{R}_{A}$, $\dot{R}_{AGS} - \dot{R}_{M}$,
$\dot{R}_{PGNCS} - \dot{R}_{AGS}$
PRINT THE ACTUAL, MEASURED, PGNCS, AND AGS RANGE AND RANGE RATE VALUES STORED
EXIT

Figure B-109.- Continued.


9
CALL CATIME TO PRODUCE A TABLE OF TIMES OF CLOSEST APPROACH OF THE VEHICLE TO THE LANDMARKS TO BE TRACKED. THE TABLE IS COMPUTED BETWEEN T_{START} AND T_{STOP}
T_{TRK} = TABLE OF CLOSEST APPROACH TIMES
N_{TOTAL} = NUMBER OF ENTRIES IN THE CLOSEST APPROACH TIME TABLE
ΔT_{TRK} = -120 SEC
$N_T = 0$
10
$N_T = N_T + 1$
$N_T : N_{TOTAL}$
>
STORE $X_{E,I}$ IN XESTAB
EXIT
≤
N:1
≤
11/9
>
$T_{TEST} = T_M(N) + \Delta T_{TEST}$
>
$T_{TEST} : T_{TRK}(N_T)$
≤

Page 8 of 10

Figure B-109.- Continued.


11
T = T_TRK (N_T) + ΔT_TRK
K = K_TRK (N_T)
R_LM = R_0 + H_K
R̄_LM = { R_LM * COS (φ_K) * COS (λ_K) ; R_LM * COS (φ_K) * SIN (λ_K) ; R_LM * SIN (φ_K) }
SIMULATE TRACKING ATTITUDE CONE
NO
[SMNB] = [I]
YES
CALL AEG TO ADVANCE X_A,I TO TIME T
CALL TRANCO TO CONVERT R̄_LM TO SELENOCENTRIC COORDINATES AT TIME T
R̄_REL = R̄_LM - R̄_A,I
CALL RRSMNB TO COMPUTE THE [SMNB] MATRIX WHICH ROTATES FROM THE PLATFORM ORIENTATION TO THE L-O-S TO THE LANDMARK
[NBSM] = [SMNB]^T
N = 0
12/10

Page 9 of 10

Figure B-109.- Continued.

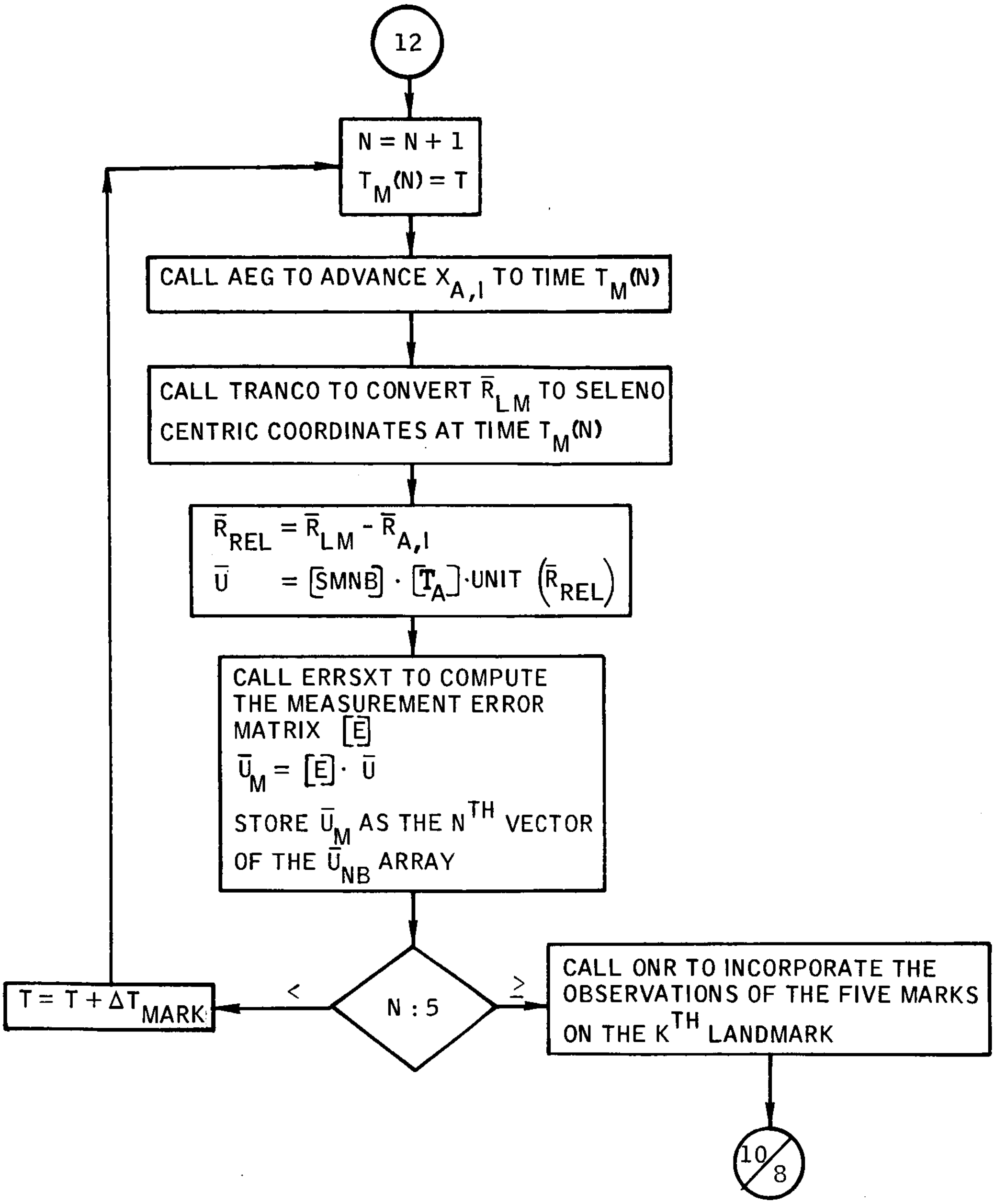
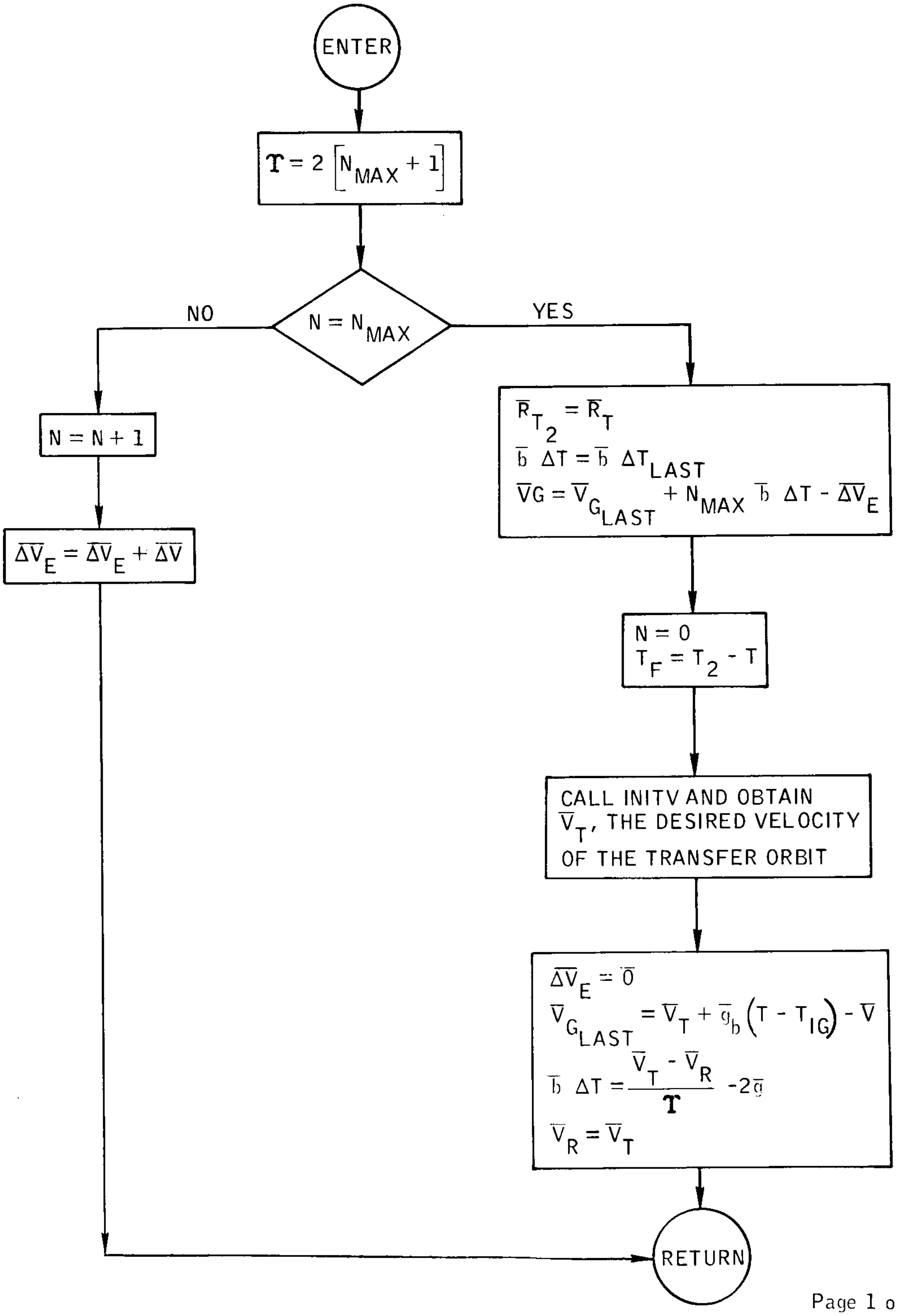

12
$N = N + 1$
$T_M(N) = T$
CALL AEG TO ADVANCE $X_{A,I}$ TO TIME $T_M(N)$
CALL TRANCO TO CONVERT $\bar{R}_{LM}$ TO SELENO CENTRIC COORDINATES AT TIME $T_M(N)$
$\bar{R}_{REL} = \bar{R}_{LM} - \bar{R}_{A,I}$
$\bar{U} = [SMNB] \cdot [T_A] \cdot \text{UNIT} (\bar{R}_{REL})$
CALL ERRSXT TO COMPUTE THE MEASUREMENT ERROR MATRIX $[E]$
$\bar{U}_M = [E] \cdot \bar{U}$
STORE $\bar{U}_M$ AS THE N^{TH} VECTOR OF THE $\bar{U}_{NB}$ ARRAY
$N : 5$
$<$
$\geq$
$T = T + \Delta T_{MARK}$
CALL ONR TO INCORPORATE THE OBSERVATIONS OF THE FIVE MARKS ON THE K^{TH} LANDMARK
10/8

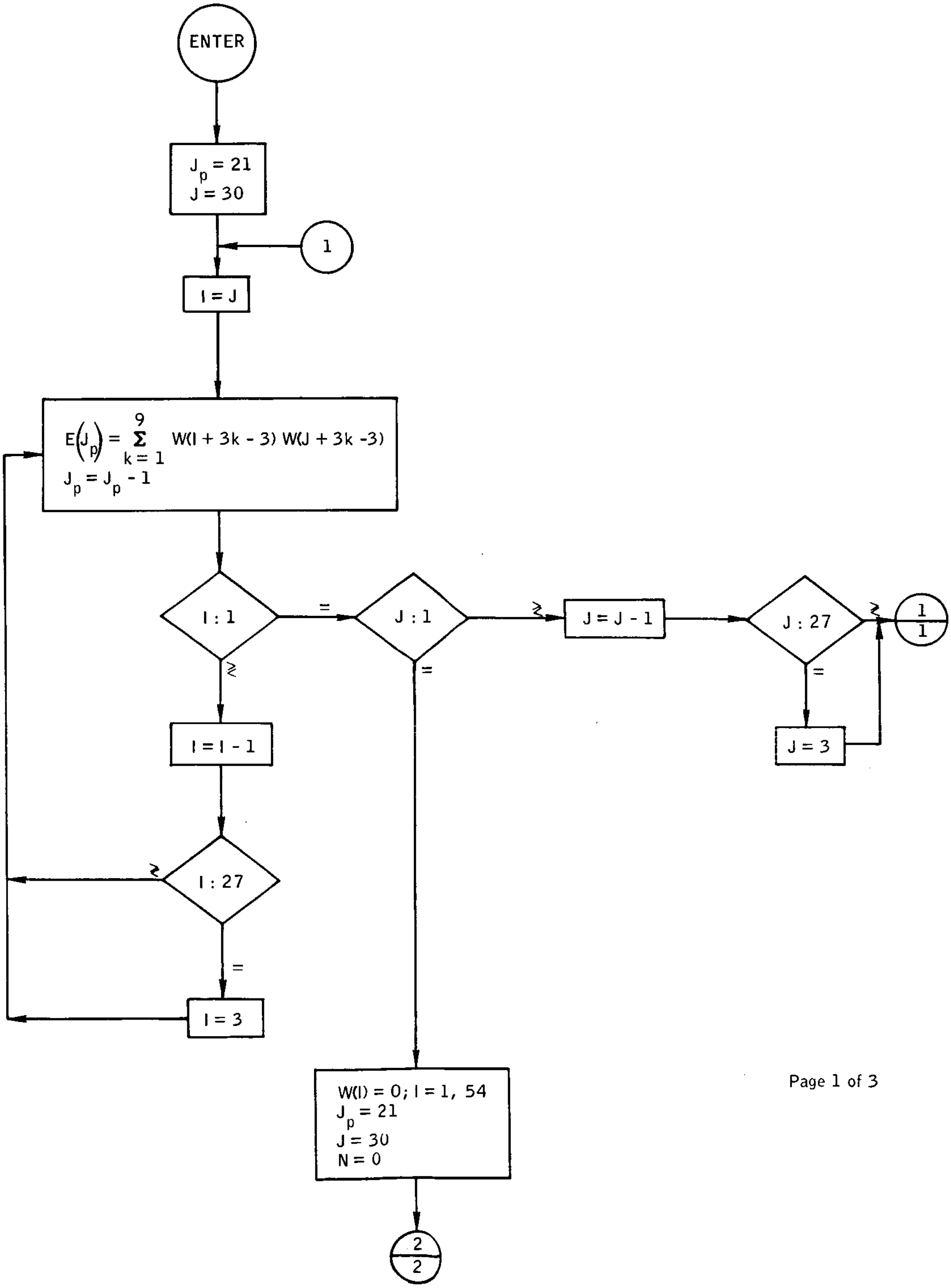
Figure B-109.- Concluded.

Page 10 of 10


ENTER
$\mathbf{T} = 2 \left[N_{MAX} + 1\right]$
$N = N_{MAX}$
NO
YES
$N = N + 1$
$\overline{\Delta V}_E = \overline{\Delta V}_E + \overline{\Delta V}$
$\bar{R}_{T_2} = \bar{R}_T$
$\bar{b}\ \Delta T = \bar{b}\ \Delta T_{LAST}$
$\bar{V}G = \bar{V}_{G_{LAST}} + N_{MAX}\ \bar{b}\ \Delta T - \overline{\Delta V}_E$
$N = 0$
$T_F = T_2 - T$
CALL INITV AND OBTAIN $\bar{V}_T$, THE DESIRED VELOCITY OF THE TRANSFER ORBIT
$\overline{\Delta V}_E = \bar{0}$
$\bar{V}_{G_{LAST}} = \bar{V}_T + \bar{g}_b\left(T - T_{IG}\right) - \bar{V}$
$\bar{b}\ \Delta T = \dfrac{\bar{V}_T - \bar{V}_R}{\mathbf{T}} - 2\bar{g}$
$\bar{V}_R = \bar{V}_T$
RETURN

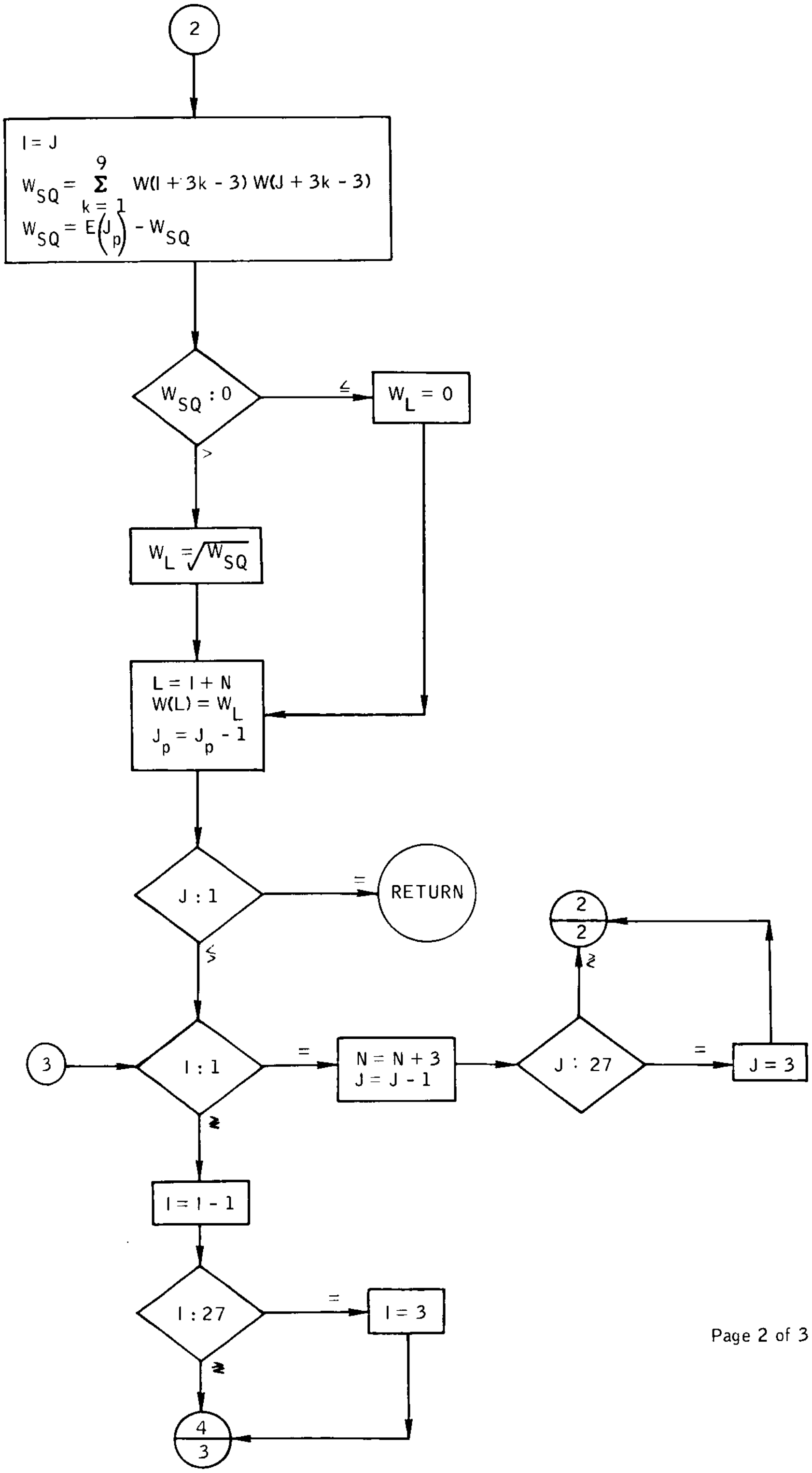
Page 1 of 1

Figure B-110.- Flow chart of subroutine VTBG.



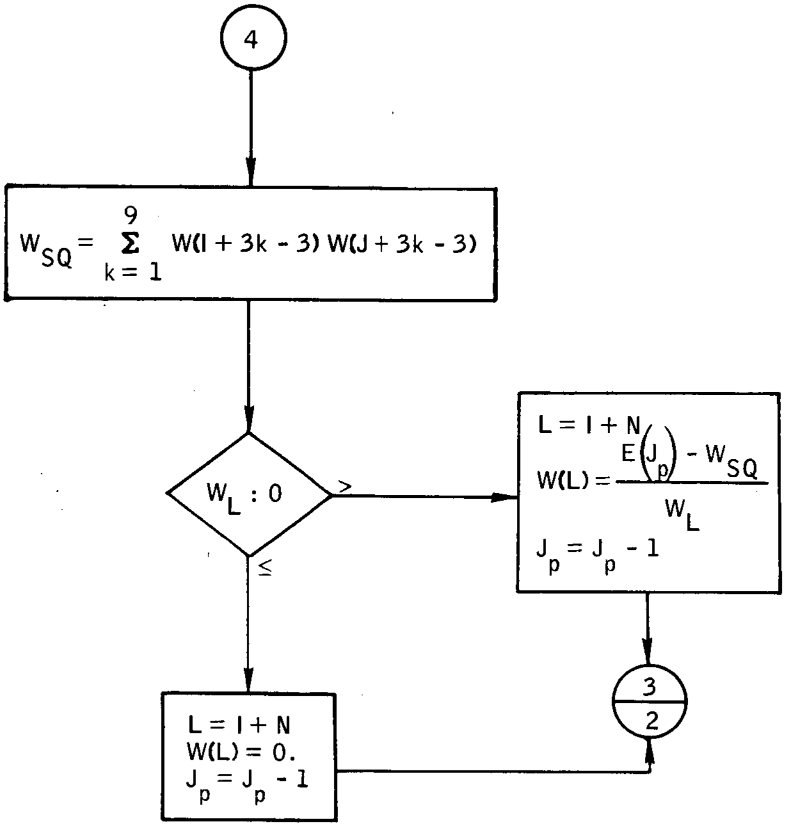
Page 1 of 3

Figure B-111.- Flow chart of subroutine WMATCN.



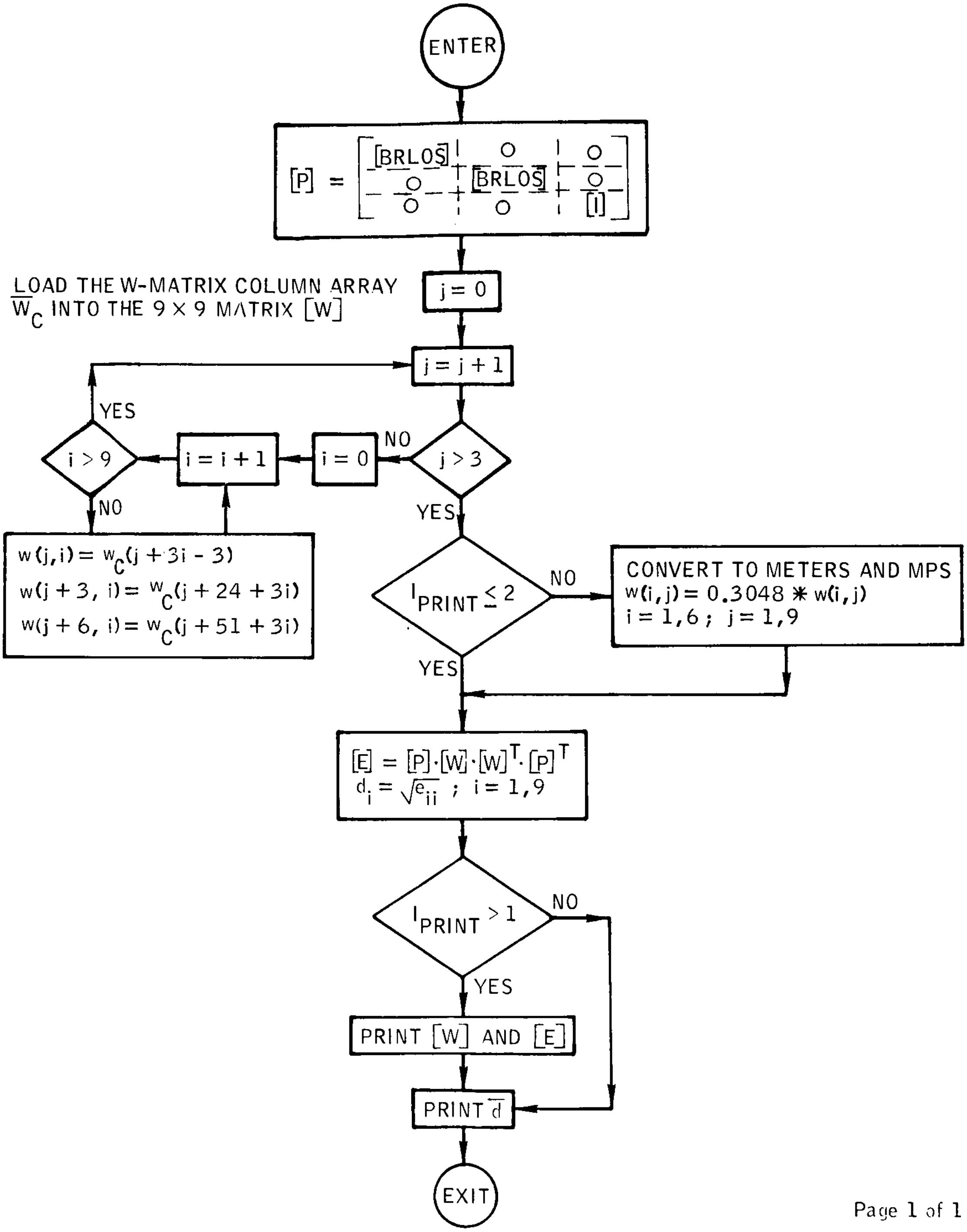
Page 2 of 3

Figure B-111.- Continued.


4
$W_{SQ} = \sum_{k=1}^{9} W(I + 3k - 3)\,W(J + 3k - 3)$
$W_L : 0$
>
≤
$L = I + N$
$W(L) = \frac{E(J_p) - W_{SQ}}{W_L}$
$J_p = J_p - 1$
$L = I + N$
$W(L) = 0.$
$J_p = J_p - 1$
3
2

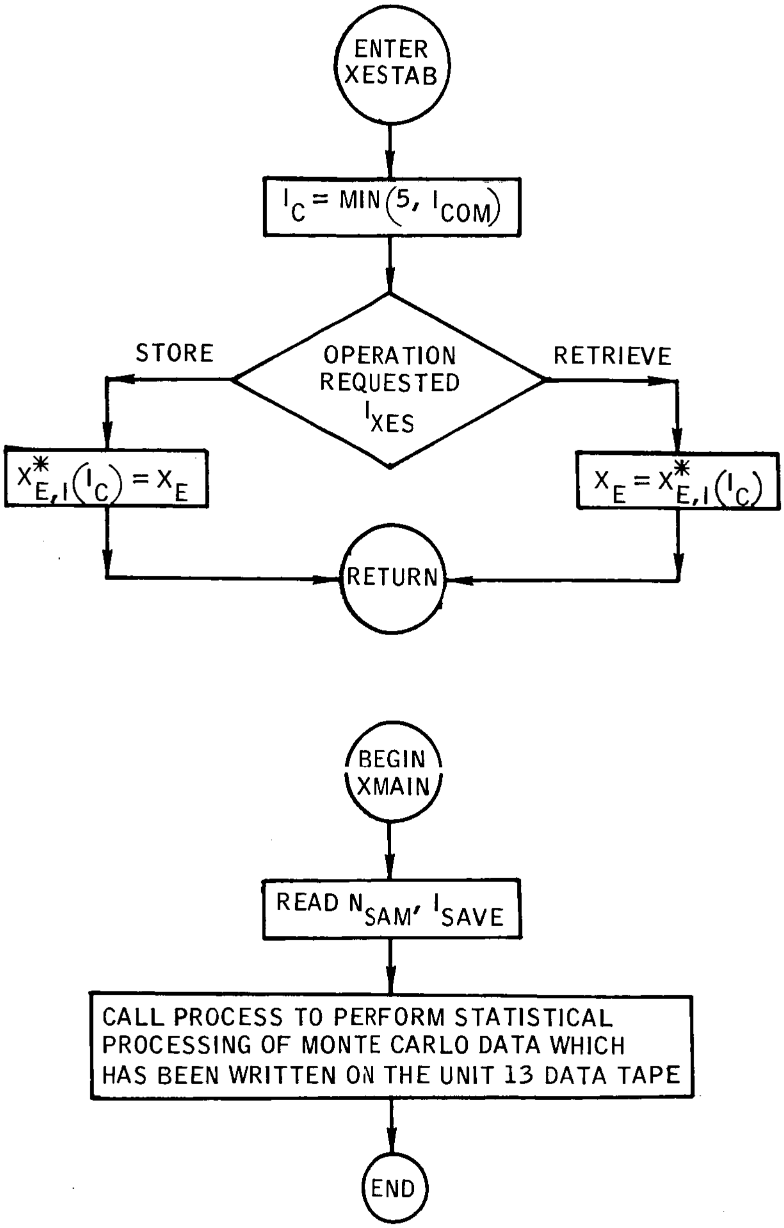
Page 3 of 3

Figure B-111.- Concluded.


ENTER
[P] = [BRLOS] | O | O ; O | [BRLOS] | O ; O | O | [I]
j = 0
LOAD THE W-MATRIX COLUMN ARRAY W_C INTO THE 9 × 9 MATRIX [W]
j = j + 1
j > 3
NO
i = 0
i = i + 1
i > 9
YES
NO
w(j,i) = w_C(j + 3i - 3)
w(j + 3, i) = w_C(j + 24 + 3i)
w(j + 6, i) = w_C(j + 51 + 3i)
YES
I_PRINT ≤ 2
NO
CONVERT TO METERS AND MPS
w(i,j) = 0.3048 * w(i,j)
i = 1,6 ; j = 1,9
YES
[E] = [P]·[W]·[W]^T·[P]^T
d_i = √e_ii ; i = 1,9
I_PRINT > 1
NO
YES
PRINT [W] AND [E]
PRINT d
EXIT

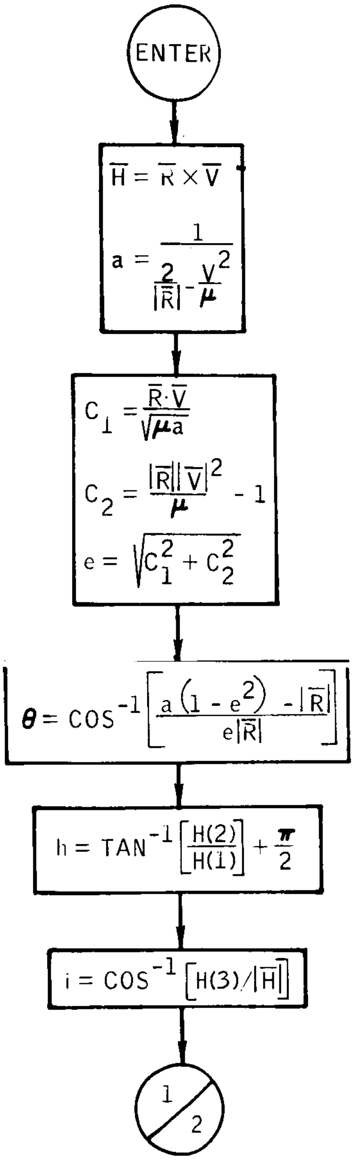
Page 1 of 1

Figure B-112.- Flow chart of subroutine WMT.


ENTER XESTAB
$I_C = MIN(5, I_{COM})$
STORE
OPERATION REQUESTED I_{XES}
RETRIEVE
$X^*_{E,I}(I_C) = X_E$
$X_E = X^*_{E,I}(I_C)$
RETURN
BEGIN XMAIN
READ N_{SAM}, I_{SAVE}
CALL PROCESS TO PERFORM STATISTICAL PROCESSING OF MONTE CARLO DATA WHICH HAS BEEN WRITTEN ON THE UNIT 13 DATA TAPE
END

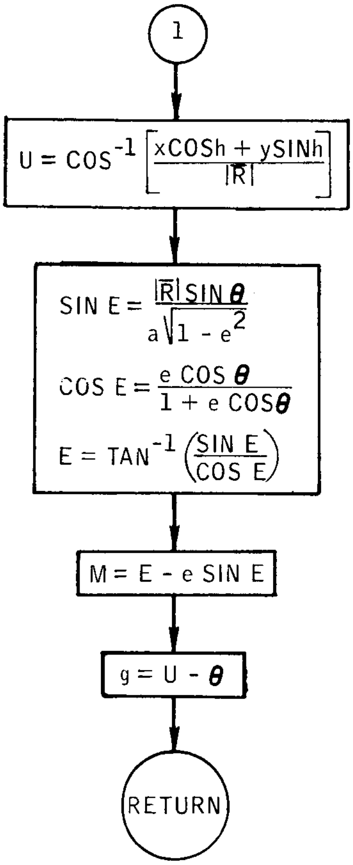
Page 1 of 1

Figure B-113.- Flow charts of subroutine XESTAB and XMAIN.


ENTER
$\overline{H} = \overline{R} \times \overline{V}$
$a = \dfrac{1}{\dfrac{2}{|\overline{R}|} - \dfrac{V^2}{\mu}}$
$C_1 = \dfrac{\overline{R} \cdot \overline{V}}{\sqrt{\mu a}}$
$C_2 = \dfrac{|\overline{R}||\overline{V}|^2}{\mu} - 1$
$e = \sqrt{C_1^2 + C_2^2}$
$\theta = \cos^{-1}\left[\dfrac{a\left(1 - e^2\right) - |\overline{R}|}{e|\overline{R}|}\right]$
$h = \tan^{-1}\left[\dfrac{H(2)}{H(1)}\right] + \dfrac{\pi}{2}$
$i = \cos^{-1}\left[H(3)/|\overline{H}|\right]$
1
2

Page 1 of 2

Figure B-114.- Flow chart of subroutine XYZTOE.


1
$U = \cos^{-1}\left[\frac{x\cos h + y\sin h}{|\bar{R}|}\right]$
$\sin E = \frac{|\bar{R}|\sin\theta}{a\sqrt{1 - e^2}}$
$\cos E = \frac{e\cos\theta}{1 + e\cos\theta}$
$E = \tan^{-1}\left(\frac{\sin E}{\cos E}\right)$
$M = E - e\sin E$
$g = U - \theta$
RETURN

Page 2 of 2

Figure B-114.- Concluded.

APPENDIX C

DEFINITIONS OF SYMBOLS USED IN SUBROUTINES OF AGAST

APPENDIX C

DEFINITIONS OF SYMBOLS USED IN SUBROUTINES OF AGAST

APS	ascent propulsion system
A_{LIM}	limiting acceleration used for rate limiting in LM descent guidance
A_{N1}	initial maneuver line number
A_{N2}	final maneuver line number
A_{TM}	magnitude of the expected acceleration vector
A_{TRIM}	prestored value of the acceleration during the trim phase of LM descent
A_{ZLIM}	limiting value on the Z component of the thrust vector
$\bar{A}_C$	commanded thrust vector in LM ascent guidance
$\bar{A}_{CG}$	total commanded acceleration in guidance coordinates; includes thrust and gravity acceleration
$\bar{A}_{CP}$	total commanded acceleration in platform coordinates
$\bar{A}_D$	acceleration vector caused by gravity
$\bar{A}_{DG}$	desired target acceleration vector in LM descent guidance
$\bar{A}_F$	unitized thrust vector, selenocentric coordinates
$\bar{A}_{FP}$	thrust vector desired, platform coordinates
$\bar{A}_T$	desired thrust vector

$[A]$ a. rotation matrix that relates the AGS strapped down platform to the AGS reference frame

b. rotation matrix from basic reference coordinates to planetary coordinates in the state vector propagation routines

$[AT]$ rotation matrix that relates the AGS strapped down platform to the BR frame

$[A_0]$ same as [AT]

a a. general symbol for semimajor axis

b. rendezvous navigation parameter in DINCP1 and DINCP2

a_I semimajor axis of the active vehicle

a_J semimajor axis of the passive vehicle

a_M magnitude of thrust acceleration in LM ascent guidance

a_N mean motion

$\bar{a}_{PERT}$ acceleration caused by gravitational perturbations of the potential

$\bar{a}_T$ thrust vector in LM ascent guidance

BR abbreviation for basic reference

BTM measured RR shaft angle

B_{DAY} base day

B_{MONTH} base month

B_{YEAR} base year

$\bar{B}_1, \bar{B}_2, \bar{B}_3$ geometry vectors used in the navigational update scheme of subroutine ONR

[B]	accelerometer misalinement rotation matrix
[BRLOS]	rotation matrix that relates the basic reference frame to a line-of-sight frame
$[BRLOS]_A$	actual [BRLOS]
[BRLV]	rotation matrix that relates the basic reference frame to a local vertical frame
$[BRLV]_A$	actual [BRLV]
$[BRLV]_E$	estimated [BRLV] based on estimated onboard states
[BRNB]	rotation matrix that relates the basic reference frame to the navigation base frame
$[B_A]$	[B]-matrix for AGS
$[B_C]$	[B]-matrix for CMC
$[B_P]$	[B]-matrix for LM PGNCS.
b	argument of latitude of closest approach to a landmark (in subroutine TPASS)
$\bar{b}\Delta t$	incremental change of the velocity-to-be-gained vector for the time interval Δt
$\bar{b}_1,\bar{b}_2,\bar{b}_3$	geometry vectors used in the navigational update routines
[b]	range or range rate RR filter weights
C	steering constant
CANT	cant angle
CART	Cartesian state vector
CARTL	Cartesian state vector (in subroutine AGSKEP)
C_E	product of eccentricity and cosine of the eccentric anomaly

$C_{\Delta E}$ cosine of the difference between the present and final eccentric anomalies

$[C]$ decomposition matrix for a covariance matrix $\sum$

$[C_{GP}]$ rotation matrix that relates the guidance frame to the platform coordinate system

c_i product of eccentricity and cosine of the eccentric anomaly (in subroutine OPS)

c_{ij} representative element of the decomposition matrix [C]

c_k input to subroutine SENSOR which designates the method used to simulate the current guidance cycle

DAYS number of days from the beginning of the year to the base date

DPS descent propulsion system

DT_{MARK} time between marks in the AGS RR update simulation

D_{CA} distance of closest approach between the CSM and LM

$\overline{D}$ vector perpendicular to the plane in which the radius, velocity, and acceleration vectors should lie at the end of the current descent phase

$\overline{D}_{IG}$ integrated gravity acceleration vector per 2-second computer cycle

E a. eccentric anomaly (subroutine TA)

b. desired elevation angle (subroutine PRETPI)

ELE classical orbital elements

E_A actual elevation angle

E_L desired elevation angle

E_{NO} number of RCS quads to be used

$[E]$	SXT unit vector error matrix
e	orbital eccentricity
F	thrust magnitude
F_A	actual thrust magnitude
F_C	commanded thrust magnitude
F_{CMIN}	minimum DPS thrust level
F_D	thrust level commanded by the descent guidance
F_{ENG}	thrust level passed to subroutine START
F_{LIM}	thrust level value used to limit the down-range componant of the thrust acceleration vector
F_{MAX}	initial DPS full thrust magnitude
F_{NOM}	nominal thrust profile
F_P	value of the commanded thrust on the previous guidance cycle
F_{SLP}	rate of change of the thrust level as a function of burn time
F_{TAB}	table of thrust magnitudes used to determine specific impulse values
F_{TRIM}	constant that is representative of the DPS thrust during the trim phase
F_1	thrust level at the beginning of each thrust phase (subroutine RUNGA)
F_2	thrust level at the end of each thrust phase (subroutine RUNGA)
F_6	flag that indicates whether or not subroutine TIMEP is to advance the state vector through the time interval T_{21}

F_{57} thrust level representative of 57 percent of DPS full-rated thrust

F_{63} thrust level representative of 63 percent of DPS full-rated thrust

F_A^* save table of actual thrust profiles

F', F'' coefficients of polynomial used to calculate LM thrust level at full throttle as a function of burn time

f coefficient used in Keplerian orbit prediction equations

$f(x)$ value of dependent variable in the regula-falsi iteration

f_3 flag set to 0 or 1 according to whether the iterator should use the regula falsi or bias method

f_4 flag set to 0 or 1 accoridng to whether the iterator is to act as a first order or a second order iterator

f_7 flag set to 1 if the input requires the conic trajectory to close through infinity

$\dot{f}$ coefficient used in Keplerian orbit prediction equations

G_{EFF} effective gravity

$\overline{G}_N$ gravity vector computed in present 2-second computer cycle

$\overline{G}$ gravity vector computed in previous 2-second computer cycle

$\overline{GRAV}$ same as $\overline{G}$

g a. acceleration caused by gravity at the earth's surface

b. argument of latitude of pericenter

c. coefficient used in Keplerian prediction equations

g_e acceleration caused by gravity at the earth's surface

g_n orbital rotation rate of the pericenter

$\dot{g}$ coefficient used in Keplerian orbit prediction equations

$\bar{g}$	acceleration vector caused by gravity
$\bar{g}_{AGS}$	same as $\overline{GRAV}$
$\bar{g}_b$	gravity acceleration perturbation caused by oblateness of the central body
$\bar{g}_p$	gravity acceleration vector used on the previous guidance cycle
H	a. altitude of a landmark above the reference sphere b. magnitude of the angular momentum vector
H_{ALT}	current altitude in the LM ascent guidance equations
H_a	altitude of the apocenter
H_k	altitude of the kth landmark above the reference sphere
H_p	altitude of the pericenter
$\bar{H}$	angular momentum vector
h	a. current altitude in the LM descent guidance equations b. argument of the ascending node of the orbital plane
h_{LS}	altitude of the landing site referenced from the mean lunar radius
h_n	rotation rate of ascending node of the orbital plane
h_{p1}	pericenter altitude on the pre-CDH orbit
h_{p2}	pericenter altitude on the pre-TPI orbit
$\bar{h}_c$	unit angular momentum vector of the CSM
I	a. index of the active vehicle b. general iteration counter

INC orbital inclination

INIT flag that indicates whether the current guidance cycle is the initial one or not

INITRR flag that indicates whether the current entry to the AGS RR logic is the first entry or not

I_{ABORT} flag that indicates whether or not one abort from powered descent is to occur

-1 nominal descent

0 an abort is to occur

1 abort has occurred

I_{AC} alarm flag that indicates the reason for convergence failure in the CSI and TPI iterations

I_{ANGLE} control flag that indicates the type of tracking data to be simulated

0 use both range and angle data

1 use range data only

2 use angle data only

I_C computer flag

I_{COM} flag that indicates the controlling computer on the current event

1 CMC

2 LGC

3 AGS

I^*_{COM} saved value of I_{COM}

I_{CON} flag that designates the units on the input reference state

I_{COUNT} general symbol for an iteration counter

I_{COV} storage/retrieval flag for the covariance matrix storage routine COVTAB

0 store

1 retrieve

I_{DIF} flag that indicates type of state vector to be processed

1 δX_A

2 δX_E

3 ΔX_A

4 ΔX_E

5 $\delta \Delta X_E$

$I_{DISPLAY}$ flag that controls print of $\Delta\Delta h$ data and closest approach data

I_{DPS} flag that indicates whether DPS engine is being used

I_{DV} flag that indicates route to be used in the subroutine LVLHDX

0 convert from LV to BR

1 convert from BR to LV

I_{ENG} flag that indicates engine to be used to perform a maneuver

1 RCS

2 DPS (or SPS for CSM)

3 APS

I_{EX} maneuver simulation control flag

0 execute and trim residuals to zero

1 execute and model trim

2 execute with no trim

3 execute with no trim and add pointing errors

4 do not execute

5 impulsive simulation

I_{FLAG} maneuver guidance control flag

1 impulsive guidance

2 external ΔV guidance

3 Lambert guidance

4 descent guidance

5 ascent guidance

6 AGS guidance

I_{FLQGO} flag that routes the logic to set the LM descent braking phase targets

I_{FLQG1} flag that routes the logic to set the LM descent approach phase targets

I_{FLVF} flag which when set to 1 routes logic to the velocity-following mode in the LM descent guidance

I_{FLVF2} flag that routes the logic to set the targets used by the velocity-following mode of the LM descent guidance

I_{FLVP} ascent vertical rise control flag

0 nominal ascent

1 vertical rise phase

I_{FPC} ascent targeting control

0 target for altitude and velocity

1 target for velocity only

I_{GUESS} regula-falsi iteration control

0 do not attempt to guess at solution until solution has been bounded

>0 make a guess at the solution even if the solution has not been bounded

I_{KNOWN} optical tracking flag that indicates whether the tracked landmark is categorized as known or unknown

I_{MAIN} index of the main thrust phase

I_{MDC} index of the current descent guidance phase

I_{ME} index of the main thrust phase

I_{MEDPS} index of the DPS main thrust phase

I_{MON} flag that indicates whether a computer is to be monitored

I_{MOVE} flag that indicates to the state vector propagation logic which time value is to be used

I_{NOMT} flag that indicates whether AGAST is currently generating a reference trajectory or a sample trajectory

I_{ONOFF} initialization flag for the guidance equations

I_{ORB} flag that indicates whether the LM is in orbit or not

I_P print flag

I_{PASS} index that indicates which solution is being computed in the targeting

1 CMC

2 LGC

3 AGS

4 MSFN

5 ACT/CMC

6 ACT/LGC

7 ACT/AGS

8 ACT/ACT

I_{PE} flag that indicates whether platform errors are to be simulated or not

I^*_{PE} saved value of I_{PE}

I_{PERTA} flag that controls the option to perturb the initial actual state vectors

I_{PERTE} flag that controls the option to perturb the initial estimated state vectors

I_{PH} index of thrust phase

I'_{PH} index of thrust phase on the previous guidance cycle

I_{PHASE} index of the current descent phase

I_{PLTN} flag that specifies whether the platform errors have been initialized or not

I_{PLTSG} flag that controls the setting of platform errors

0 use input errors

1 compute platform errors from σ's

I_{PRINT} print flag

I_R flag that designates how the AEG is to output U and M

0 output U and M in the intervaul $[0, 2\pi]$

1 output U and M in the proper revolution

I_{RAN} controls the logic routing of the random number generator

1 output zeros only

0 initial call to RANNO

1 subsequent call to RANNO

I_{RBIAS} RR bias control flag

I_{ROT} indicates whether an input covariance matrix is to be rotated or not

I_{SAM} flag that indicates whether a mission event is to be a reference point and what kind of statistical processing is to be performed

I_{SAVE} reference point counter

I_{SET} indicates whether the input vector is in classical, Cartesian, or spherical elements

I_{SP} specific impulse

I_{SP_A} acutal specific impulse of the current thrust phase

$I^*_{SP_A}$ stored values of specific impulse for each thrust phase

I_{SP_D} specific impulse that corresponds to the commanded DPS thrust level

$I_{SP_{NOM}}$ nominal specific impulse

I'_{SP}, I''_{SP} first and second degree time derivatives of the specific impulse used in the polynomial in the descent guidance

I_{SPNP} specific impulse of the tail-off phase

I_{SPLIT} flag that indicates if the burn ignition time and the ΔV of an external ΔV maneuver are to be biased by subroutine SPLIT to account for the finite burn effects

0 do not bias the burn

1 bias the burn

I_{SPTAB} table of specific impulse values used to calculate DPS specific impulse as a function of the commanded thrust

I_{STAGE} flag that defines the status of the DPS stage during powered ascent

1 nominal ascent with APS or DPS ascent without staging the DPS

0 DPS staging is to occur

1 DPS staging is to occur on the current guidance cycle

2 staging has occured ond the burn is being completed with the APS

I_{STATE} flag set to 1 or 0 depending upon whether or not the initial state vectors for ascent have been computed by AFLPN

I_{STBRN} flag that initializes the T_{GO} calculations by calling SHORTB

I_{STOP} flag to control the logic which sets the time to end onboard tracking

I_{SXTB} SXT bias control flag

I_{TDV} flag set to 0 or 1 depending upon whether or not the automatic target ΔV option is desired on all monitored computers after a burn

I_{TEV} flag that controls the setting of an event time

I_{THR} flag to control the logic which sets up a model of the actual thrust profile

I^*_{THR} flag for each engine which designates whether a constant or linearly varying thrust model is to be used for the main engine phase of a burn

I_{THROT} flag that indicates if the DPS engine is to throttle up despite the short burn logic constraints

I_{TIME}	flag that indicates the form of the input reference state time
I_{TYPE}	flag that indicates whether an input covariance matrix represents actual state vector uncertainties or estimated state vector uncertainties
I_{UPDATE}	flag that indicates if the state vectors which result from a MSFN update are to be transfered to the monitoring computers
I_V	vehicle flag used in subroutine PERRO
I_{VEH}	index of active vehicle; in subroutine ONR, it indicates the state vector being updated
I_{VR}	vehicle index of the state vector which is to be transferred from one computer to update another computer
I_{WMATRX}	W-matrix initialization control flag
I_{WRITE}	flag that indicates if the out-of-plane velocity at a maneuver point is to be nulled
I_{2PH}	flag that indicates whether or not two-phase descent is to be used
$[I]$	the n by n identity matrix
i	orbital inclination
J	index of passive vehicle
J_D	dimension of the W-matrix (may be either 6 by 6 or 9 by 9)
J_{DFLAG}	flags that indicate which of the maneuver targeting differences are to be produced
J_{DGZ}	the desired Z-component of the target jerk vector in the LM descent guidance
J_{ENTER}	flag that indicates a reentry condition in ENCKE
J_F	routing flag for the print logic of the onboard tracking routines

J_{FLAG} a. internal routing flag for the logic of subroutine RNR

b. in descent, this flag determines which set of target vectors will be used for the current phase

$J_{KT,PHASE}$ the Z-component of the target jerk vector for each phase of LM descent

J_{LOOP} flag that controls the cycle iteration during the preignition phase

J_{MFLAG} flag that designates which sections of the LM ascent insertion covariance matrix are to be entered as input

J_{NEXT} flag that indicates whether or not a T_{NEXT} time is to be computed in TARGET

J_{PASS} first value of I_{PASS} in the maneuver targeting loop in subroutine TARGET

J_{PHASE} flag that indicates current descent guidance phase

1 braking phase

2 approach phase

J_{SFLAG} flags that designate which of the maneuver targeting solutions are desired

J_{SKIP} increment by which I_{PASS} is increased in TARGET after each targeting solution has been obtained

J_V flag that indicates which vehicle's state vector is to propagate the W-matrix in ENCKE

J_1 index input to SETTHR that indicates the first engine to be modeled in the thrust setting logic

J_1^{28} component of external ΔV in down-range direction

J_2 a. index input to SETTHR that indicates the last engine to be modeled in the thrust setting logic

b. in the state vector propagation routines, J_2 designates a constant in the gravitational potential function

J_2^{28}	component of external ΔV in cross-range direction
J_{2E}	the J_2 constant for the earth gravitational potential function
J_3	constant in the gravitational potential function
J_3^{28}	component of external ΔV in negative radial direction
J_4, J_{22}, J_{31}	constant in gravitational potential function
J_{1900}	Julian date of January 1, 1900
J_{1964}	Julian date of January 1, 1964
J^1	desired TPI time
J^2	cotangent of desired LOS angle angle at TPI time
J^5	nominal landing site radius
J^6	desired LM transfer time for direct intercept transfer routine
J^7	term in LM semimajor axis α_L in orbit insertion route
J^8	lower limit of α_L in orbit insertion route
J^9	upper limit of α_L in orbit insertion route
J^{16}	orbit insertion targeted injection altitude
J^{17}	radar range rate
J^{18}	radar range
J^{21}	vertical pitch steering altitude threshold
J^{22}	vertical pitch steering altitude rate threshold
J^{23}	orbit insertion targeted injection radial rate

J^{25} altitude above lunar surface for an altitude update

J^{26} radar filter update time initialization value

K a. in subroutine SAMSIG, this flag indicates the number of variables to be processed

b. in subroutine COMDKI, this flag is an iteration counter

KA AGSKEP initialization flag

KI initialization flag in subroutine AGSKEP

KORIN flag that indicates the coordinate system of a state vector being entered into a propagation routine

KOROUT flag that indicates the desired coordinate system of a state vector to be output from a propagation routine

KRNDFG flag that indicates whether or not the onboard tracking equations have been initialized

K_{BS} flag that indicates whether or not the tracking vehicle is to boresight along the line of sight

K_D a. in AMAIN and INCOV, this parameter is the index of a covariance matrix

b. in the tracking incorporation routines, this parameter is the index of the update matrix

K_{DIR} flag used to determine the thrust direction in the ascent guidance

K_{ENTER} routing option for subroutine START

K_{EPOCH} initialization parameter for JPLEPH

K_{EST} flag that indicates whether the estimated states are to be propagated with the AEG or with the onboard integrator

K_{EVENT} event counter

$\{K_i\}_{i=0}^{3}$ powered descent abort constants

K_{MODE} flag that indicates whether or not subroutine SAMPLE is to output a perturbed state vector X

K_{NSR} iteration counter in subroutine JOCDH

K_{STBRN} flag that indicates if the burn is less than 6 seconds in length

K_{STEER} flag that indicates if steering is allowed on the current guidance cycle

K_{STBURN} same as K_{STBRN}

K_{S10} guidance selection flag

0 orbit insertion

1 CSI

2 CDH

3 TPI search

4 TPI execute

5 external ΔV

K_{TARG} flag that controls the routing logic of DKI maneuver computations

K_{TGO} flag in the ascent guidance equations which prohibits the cycling of the T_{GO} equation after T_{GO} becomes less than 4 seconds

K_{THR} flag that indicates whether or not the thrust model is to be changed for the current maneuver event

K_{TOFF} flag that indicates whether the burn is in the tail-off phase or is complete

K_{TRK} code associated with the T_{TRK} table and consisting of a table of indicies for the landmarks to be tracked

K_{UL} ullage phase indicator

K'_{UL} value of K_{UL} on previous cycle through the integrator

K_{UTA} flag that controls the routing of the platform error model logic of subroutine MANEUV

K_{VR} vertical rise indicator in the LM ascent guidance logic

0 steer in the commanded direction

1 steer along the radius vector

K_X coefficient of the x term of the equation used to solve for powered descent initiation

K_Y coefficient of the y term of the equation used to solve for powered descent initiation

K_V coefficient of the velocity term of the equation used to solve for powered descent initiation

K_{XTRA} counter which indicates how many variables have been loaded into the XTRA array

K_1 a. initialization flag for the state vector propagation routines

b. index input to SETTHR which indicates the first vehicle to be modeled in the thrust setting logic

c. engine constant in subroutine SHORTB

K_2 a. index input to SETTHR which indicates the last vehicle to be modeled in the thrust setting logic

b. engine constant in subroutine SHORTB

K_6 counter that indicates the index of the 6 by 6 covariance matrix to be used on the current event

K_{12} counter that indicates the index of the 12 by 12 covariance matrix to be used on the current event

K_1^2 AGS gravitational constant

K_2^2 reciprocal of $K_1{}^2$

K_3^2	value of g in AGS if overflow occurs in e^2
K_4^2	$-K_1^2\Delta t$ ($\Delta t = 2$ sec)
K_4^4	coefficient in linear expression for $\dot{r}_f$ as a function of r_f
K_5^4	constant in linear expression for $\dot{r}_f$ as a function of r_f
K_6^4	upper limit on $\dot{r}_f$
K_{11}^2	value to which V_T is set if overflow occurs in V_T
K_{12}^4	acceleration check for $\dddot{R}_{DTL}$
K_{14}^5	upper limit on $\dddot{r}_d$
K_{17}^2	controls iterations of p-iterator
K_{17}^5	lower limit on $\dddot{y}_d$
K_{18}^5	lower limit of r_d
K_{19}^2	Δp limiter in p-iterator
K_{20}^2	p-iterator convergence check
K_{20}^5	lower limit of $\dddot{r}_d$
K_{26}^5	velocity-to-be-gained threshold
$\bar{K}_{XL}$	constants associated with the landmark and used in landmark tracking
k_Q	flag that indicates the type of rendezvous navigation parameter for which print is desired
L	a. vehicle index
	b. intermediate matrix for range and range rate updates in AGS

LM_{UP} rendezvous navigation flag that indicates whether the state vector being updated is that of the LM or that of the CSM

LOS line of sight

LV local vertical

L_{EC} primary flag for event category

1 maneuver

2 platform alinement

3 update event (using a covariance matrix)

4 update event (model onboard tracking)

L_{EN} secondary flag for event category; e.g., this flag is used to differentiate between the different maneuver types

L_{SA} flag that indicates how the actual landing site position is to be calculated

L_{SE} flag that indicates how the estimated landing site position is to be calculated

L_1 index that indicates the first state vector to be propagated in the AEG and in ENCKE

L_2 index that indicates the last state vector to be propagated in the AEG and in ENCKE

[LVBR] rotation matrix that relates the Apollo local vertical frame to the basic reference frame

$[LVBR]_A$ actual [LVBR]

[LVLOS] rotation matrix that relates the Apollo local vertical frame to the rendezvous navigation line-of-sight frame

$[LVLOS]_A$ actual [LVLOS]

ℓ_{MAX} upper bound of general independent variable

ℓ_{MIN} lower bound of general independent variable

ℓ_1,ℓ_2 indices that indicate which computer platforms are to be modeled in SETPLT

M mean anomaly

M_{AGS} burn monitor flag for the AGS computer

M_0 array that contains the number of days in each month of the year

M_{PGNCS} burn monitor flag for the LGC and CMC computers

M_{PT} counter that indicates the number of maneuvers stored in the mission plan table

m

a. vehicle mass

b. matrix of partial derivatives of the computed range rate with respect to inplane position and velocity component

$\dot{m}$ vehicular mass flow

N

a. mark counter for the tracking routines

b. number of Monte Carlo samples in the statistical processing routines

c. counter that indicates the number of guidance cycles which have passed since the last Lambert solution in a Lambert guided maneuver

N_{DEL}

0 for range rate radar update in AGS

1 for range radar update in AGS

N1 flag that indicates the number of offsets to be computed in subroutine INITV

N_c counter used in the thrust sensing logic in MANEUV

N_{CASE} Monte Carlo loop counter

N'_{CASE} counter that indicates number of cycles of the AGAST logic, including the cycle for generation of the reference trajectory

N_{CDH} apsis indicator for the CDH maneuver

N_F maneuver line counter to be advanced to in subroutine COMDKI

N_I initial maneuver line counter in subroutine COMDKI

N_{LM} number of landmarks to be tracked

N_{MAX} number of guidance cycles in a Lambert guided maneuver between each Lambert solution

N_P index of the tail-off phase

N_{PASS} iteration counter for the T_{GO} calculation of the descent guidance

N_{PH} burn simulator control flag that keeps I'_{PH} from being set until all logic required for initialization of a new thrust phase has been processed

N_{SAM} number of Monte Carlo cycles requested

N_{THROT} flag that indicates whether or not the DPS is to be throttled up

N_{TOTAL} counter for the T_{TRK} table

N_1 a. maneuver line number of the first maneuver of a DKI sequence

b. otherwise equivalent to N_1

N_2 a. maneuver line number of the second maneuver of a DKI sequence

b. otherwise equivalent to N_F

N_3 a. maneuver line number of the third maneuver of a DKI sequence

b. in subroutine START, this parameter is the index of the main thrust phase

N_4 index of the main engine ignition thrust phase in subroutine START

N_5 index of the tail-off thrust phase in subroutine START

[NBSM] rotation matrix that relates the navigation base coordinate frame to the stable member (IMU) frame

n variable that indicates which actual solution is to be used for computation of the maneuver targeting error

n_{CSM} mean motion of the CSM

n_E mean motion of CSM in AGS

n_I mean motion of the ith vehicle

n_i mean motion output by orbit parameters subroutine

P
a. orbital period in subroutine COMDKI

b. LM state vector error covariance matrix in AGS

c. semilatus rectum of LM transfer orbit in AGS

PDI
a. powered descent initiation

b. time of powered descent initiation

PEM platform error matrix in subroutine AGSRR

P_F thrust dispersion percentage desired

P_{HT} equivalent to ΔT_{BP}

P_I specific impulse dispersion percentage desired

P_N ratio of the semilatus rectum to the position vector magnitude

P' previous value of p in p-iterator portion of AGS

[P] rotation matrix output from subroutine PMATRX

$[P(t)]$ platform error matrix

$[P_A(t)]$ AGS platform error matrix

$[P_E(t)]$ platform error matrix output from subroutine PERRO

$[PNL]$ lunar libration matrix which is output from subroutine JPLEPH; see the second definition of [A]

$[P_T]$ PGNCS platform error matrix at the time of AGS-to-PGNCS alinement

p_i probability table generated in the cumulative distribution function routine CDFUNC

Q intermediate computation matrix for radar update equations

q present LM perigee

q_a same as q

q_1 perigee of LM transfer orbit

R radius vector magnitude

RR rendezvous radar

R_A a. actual relative range in tracking routines

b. radius of apogee in CONVRT

R_{AGS} estimated relative range in the AGS

R_a radius of apogee

R_{BIAS} bias to RR range measurement

R_{DD} desired LM insertion radius

R_E earth equatorial radius magnitude

R_{EPH} conversion factor from er/hr to fps

R_{IGX},R_{IGZ}	desired x and z components of the LM position vector at the full throttle point of descent
R_J	target radius vector magnitude
R_L	chaser radius vector magnitude
R_{LS}	radius magnitude of the landing site
R_{LSM}	same as R_{LS}
R_M	measured relative range
R_{MOON}	mean lunar radius
R_O	equivalent to R_{MOON}
R_P	radius of perigee
R_{PGNCS}	estimated relative range in the PGNCS
R_{REF}	mean radius of the central body
R_{TA}	CSM state vector in the AGS
R_x,R_z	components of $\bar{R}$ along local vertical and local horizontal axes, respectively
$\dot{R}$	a. radical velocity component
	b. relative range rate
$\dot{R}_A$	actual relative range rate
$\dot{R}_{AGS}$	estimated relative range rate in the AGS
$\dot{R}_M$	measured relative range rate
$\dot{R}_{PGNCS}$	estimated relative range rate in the PGNCS

$\dot{R}^{*}$ computed range rate saved when subroutine FILTER is initialized

R_{DTL} lower limit of r_d, computed as a function of vehicle configuration

$\bar{R}$ a. position vector

b. relative position vector of CSM with respect to the LM

$\bar{R}^{**}$ computed inertial range vector in subroutine FILTER

$\bar{R}_{AGS}$ AGS estimated LM position vector

$\bar{R}_{A,I}$ actual position vector of the active vehicle

$\bar{R}_{A,J}$ actual position vector of the passive vehicle

$\bar{R}_{AVG}$ position vector to be integrated by subroutine AGSAG (input and output)

$\bar{R}_{C}$ position vector used in the ascent guidance

$\bar{R}_{CL}$ range vector directed from the CSM to a landmark

$\bar{R}_{CON}$ conic position vector in ENCKE

$\bar{R}_{CSM}$ position vector of the CSM

$\bar{R}_{DG}$ desired target position vector

$\bar{R}_{E,I}$ estimated position vector of the active vehicle

$\bar{R}_{E,J}$ estimated position vector of the passive vehicle

$\bar{R}_{G}$ position vector in LM descent guidance coordinates

$\bar{R}_{I}$ position vector of the active vehicle

$\bar{R}_{IG}$ position vector at ignition time

$\bar{R}_{I1}$	position vector of the active vehicle at CSI
$\bar{R}_{I2}$	position vector of the active vehicle at CDH
$\bar{R}_{J}$	position vector of the passive vehicle
$\bar{R}_{JC}$	passive vehicle position at CDH phase match
$\bar{R}_{J1}$	position vector of passive vehicle at CSI
$\bar{R}_{J2}$	position vector of passive vehicle at CDH
$\bar{R}_{J3}$	position vector of passive vehicle at TPI
$\bar{R}'_{J}$	position vector of the passive vehicle at phase match with the desired position of the chaser at TPI
$\bar{R}_{L}$	a. position vector of a landmark in subroutine ONR b. magnitude of position vector in TCONE
$\bar{R}_{LC}$	vector from the active vehicle to the passive vehicle
$\bar{R}_{LM}$	position vector of the LM
$\bar{R}_{LS}$	landing site position vector
$\bar{R}_{LSA}$	actual landing site position vector
$\bar{R}_{M}$	position vector at the midpoint of a burn
$\bar{R}_{N}$	new value of position vector integrated by subroutine AGSAG
$\bar{R}_{0}$	a. position vector input to subroutines LVLHDX and PMATRX b. rectified state vector in ENCKE
$\bar{R}^{*}_{0}$	saved value of $\bar{R}_{0}$ in ENCKE

$\bar{R}_P$ a. position vector in platform coordinates in the LM descent guidance

b. landmark position vector in selenocentric coordinates in subroutine ONR

$\bar{R}_{PGS}$ PGNCS estimated LM position vector

$\bar{R}_{PP}$ range vector in platform coordinates directed from the LM to the landing site

$\bar{R}_{REL}$ actual relative range vector

$\bar{R}_S$ position vector of the landmark to be tracked

$\bar{R}_{SG}$ landmark position vector in selenographic coordinates

$\bar{R}_T$ target vector for Lambert maneuver

$\bar{R}_{T,JPHASE}$ input target position vector for each descent guidance phase

$\bar{R}_{T2}$ offset target vector in a Lambert maneuver

$\dot{\bar{R}}$ computed range rate vector between LM and CSM

$[R]$ a. see $[P]$

b. rotation matrix from first point of Aries to Greenwich coordinates in AMATRX

$[ROT_{RR}]$ matrix that defines the actual orientation of the LM Z-body axis

$[R_0]$ matrix that defines the vehicle attitude during a burn

r a. position vector magnitude

b. range in PCA

r_b predicted LM burnout radius saved from previous 2-second cycle

r_f predicted LM radial distance at t_{ig}, current for orbit insertion

r_i inertial position vector magnitude output by subroutine OPS

r_T predicted radial distance of CSM as it crosses the vector $\bar{r}_5$ in the direction of the LM at t_{ig}

$\dot{r}$ range rate

$\dot{r}_A$ predicted LM altitude rate at t_{ig} for CSI and CDH

$\dot{r}_f$ predicted radial rate for the following

a. CSM as it crossed the vector $\bar{r}_s$ in the direction of the LM at t_{ig} for CDH

b. LM in orbit insertion mode

c. LM at t_{ig} for CSI

$\ddot{r}_d$ desired radial acceleration

$\dddot{r}_d$ desired derivative of the radial acceleration

$\bar{r}$ a. LM position vector

b. position vector to be advanced by subroutine CONIC

$\bar{r}_C$ predicted CSM inertial position vector at t_{ig} for CSI, CDH, or TPI modes; at present time for orbit insertion mode

$\bar{r}_0$ position vector input to subroutines OPS and ELPRE

$\bar{r}_5$ predicted LM inertial position vector at t_{ig}; current LM position vector for orbit insertion

$\dot{\bar{r}}_F$ LM velocity vector just prior to the second impulse in the transfer orbit

$\dot{\bar{r}}_I$ LM velocity vector after first impulse for the transfer orbit

S intermediate symbol for a sum

SAVMAN array that contains the following data for each solution targeted

a. T_{EX}

b. ΔV_{EX}

c. $\dot{y}$

d. h_a

e. h_p

f. E_A

SOM stable orbit midcourse maneuver

SOR stable orbit rendezvous maneuver

SPHERE six-dimensional array of spherical orbital elements

S_E product of the orbital eccentricity and the sine of the eccentric anomaly

S_i product of eccentricity and sine of eccentric anomaly

$S_{\Delta E}$ sine of the difference in the present and final eccentric anomalies

S_{07} 1 if velocity-to-be-gained vector is to be frozen in AGS inertial coordinates and sensed velocity;

0 otherwise

S_{10} same as K_{S10}

S_{16} 1 compute CDH at one-half LM orbital period after CSI

3 compute CDH at 1.5 LM orbital periods after CSI

[SMNB] rotation matrix that relates the stable member (IMU) frame to the navigation base coordinate frame

S_j accelerometer scale factor in subroutine SETPLT

T current mission time referenced from the base time

TH_{CL} threshold time for the maneuver line logic

TIME a. clock time read from the UNIVAC 1108 system clock

b. time to which a state vector is to be advanced in ENCKE

T_{A} time of alinement

T_{ABORT} time at which an abort from descent is to occur

T_{AGS} time associated with the AGS state vector

T_{AO} time from CSI to CDH in AGS

T_{ASCENT} nominal ascent powered flight time

T_{B} time of burn

T_{BIAS} time bias added to the CDH or TPI maneuver time

T_{CA} time of closest approach of the LM and CSM

T_{CDH} time of the CDH maneuver

T_{CSI} time of the CSI maneuver

T_{COMP} time between each Lambert solution in a Lambert guided burn

T_{D} direct transfer time through which the conic update of the state vector is to be made

T_{DV} time used in the target ΔV logic

T_{E} time associated with an estimated state vector

T_{E}^{*} saved value of T_{E}

T_{ENDAPR} the ΔT, referenced from the LM descent approach phase target point, at which the approach phase is terminated

T_{ENDBRK} the ΔT, referenced from the LM descent braking phase target point, at which the braking phase is terminated

T_{ERR} error in the transfer time in subroutine LAMBERT

T_{EVENT} time of the current event

T_{EX} time of an external ΔV maneuver

T_F
a. time of the final maneuver in a Lambert sequence
b. mark time in the optical tracking routines

T_{FA} time of the first apsis crossing after a threshold time

T_{FTP} time at which DPS full throttle position is to be achieved

T_{GO}
a. in the LM descent guidance logic, this parameter represents the time to go until the end of the current descent guidance phase
b. in all other guidance logic, this parameter represents the time to go until the main engine cut off.

T'_{GO} time to go until end of tail-off

T''_{GO} time to go until end of the current thrust phase

T^*_{GO} stored T_{GO} which was calculated during the current guidance cycle

T_{GOAUG} first guess at the approach phase duration

T_I time of the first maneuver in a Lambert sequence

T_{IG} time of burn initiation

T_{IM} table of the ΔT's of each thrust phase used in subroutine START

T_{IMN4} ullage phase ΔT

T_i time input to ELPRE which represents one of the following
a. time from CSM epoch to rendezvous
b. time from CSM epoch to rendezvous retarded by J^4 in TPI mode

c. time from CSM epoch to CSM crossing vector in $\bar{r}_5$ direction in CSI and CDH modes

T_{ig} desired time of next AGS maneuver

T_{iga} time of CSI in AGS

T_{igb} time of CDH in AGS

T_{JC} time in Julian centuries from January 1, 1900, to the base date

T_{JD} Julian date of the base date

T_{LAST} a. time of the last event

b. time of the last initial state vector in the AEG

T_{LO} lift-off time

T_{M} table of five mark times for a pass over a landmark

T_{MARK} vector of tracking sequence times in subroutine AGSRR

T_{MOVE} time to which the state vectors are to be propagated

T_{NEXT} time of the next event

T_{NEXTE} estimated T_{NEST} values for each monitored computer

T_{NSR} time of the NSR maneuver

T_{N1} time of the first maneuver of a DKI sequence

T_{N2} time of the second maneuver of a DKI sequence

T_{N3} time of the third maneuver of a DKI sequence

T_{P} a. orbital period

b. time from TPI to rendezvous

T_P' previous value of T_p in p-iterator portion of AGS

T_{perg} time of LM to perilune

T_R a. time of the initial reference state vector

b. predicted time from present to rendezvous (valid only for TPI mode)

T_{SAVE} table of closest approach times of the CSM a landmark

T_{SEC} time to which a state vector is to be advanced

T_{SEC}^* saved value of T_{SEC}

T_{START} start time for onboard tracking

T_{STOP} stop time for onboard tracking

T_T time at which the passive vehicle is at phase match with the desired SOR target

T_{TPF} time of rendezvous

T_{TPI} time of terminal phase initiation

T_{TRK} table of closest approach times of the CSM to a set of landmarks

T_{XUS} average-g integration time interval left before burn ignition

T_Δ time from present to next maneuver

T_1 a. time of the first maneuver of a Lambert sequence

b. time of last radar range update

T_2 time of the second maneuver of a Lambert sequence

T_4 time since the DPS went to the full throttle position

T_{21} computed transfer time from Kepler's equation

T'_{21}	previous value of T_{21}
T'	time of the last guidance cycle
$[T]$	REFSMMAT; rotation matrix that relates the basic reference system to the estimated IMU orientation
$[T_A]$	rotation matrix that relates the basic reference system to the actual IMU orientation
t	time of LM state vector in AGS
t_E	time of CSM state vector in AGS
t_{ig}	desired time of next maneuver
t'_{21}	previous value of t_{21} in iteration
U	vehicle's argument of latitude
U_{AP}	argument of latitude of the apoapsis
U_{CSI}	argument of latitude of the CSI maneuver
U_I	argument of latitude of the active vehicle
U_J	argument of latitude of the passive vehicle
U_{OC}	argument of latitude of the maneuver line
U_{1X}	x-component of vector $\bar{U}_1$
U_{1Z}	Z-component of vector $\bar{U}_1$
U^*	saved value at U
$\bar{U}$	general symbol for a unit vector
$\bar{U}_B$	burn residual
$\bar{U}_{CL}$	unit vector along $\bar{R}_{CL}$
$\bar{U}_M$	unit vector along the line of sight of the SXT to an object

$\bar{U}_{NB}$ SXT navigation base unit vector directed along the line of sight to an object being tracked

$\bar{U}_{Q}$ unit vector in the LM ascent guidance along the negative CSM angular momentum vector

$\bar{U}_{R}$ unit vector along the radius vector

$\bar{U}_{REL}$ unit vector along the relative range vector

$\overline{U}_{r_1}$ unit vector of $\bar{r}_1$ in subroutine CONIC

$\bar{U}_{SM}$ unit vector in stable member coordinates along the line of sight to a landmark being tracked

$\bar{U}_{SXT}$ unit vector produced by the alinement of the SXT to in object to be tracked

$\bar{U}_{TA}$ unit vector along the actual thrust direction

$\bar{U}_{TD}$ unit vector along the desired thrust direction

$\bar{U}_{y}$ unit vector in the LM ascent guidance which is perpendicular to the LM radius vector

$\bar{U}_{Z}$ down-range unit vector at LM insertion

$\bar{U}_{ZR}$ vector through the central body north pole

$\bar{U}_{1}$ unit LM position vector

$\bar{U}_{2}$ CSI and CDH: unit vector pointing toward the CSM at t_{ig}

TPI: unit vector pointing toward either rendezvous point or toward rendezvous point retarded by J^4

[UVWXYZ] rotation matrix that relates the uvw local vertical coordinate system to the xyz Apollo local vertical coordinate system

V general symbol for velocity

VAR_{IMU} onboard constant that represents the a priori estimate of the IMU angular error variances per axis

VAR_{INT} onboard constant that represents the a priori estimate of the relative range error variance caused by coasting integration inaccuracies

VAR_{R} onboard constant that represents the a priori estimate of the RR measured range error variance corresponding to a percentage error

VAR_{RMIN} onboard constant that represents the minimum RR measured range error variance

VAR_{SCT} onboard constant that represents the a priori estimate of the scanning telescope angular error variance per axis

VAR_{SXT} onboard constant that represents the a priori estimate of the sextant angular error variance per axis

VAR_{V} onboard constant that represents the a priori estimate of the RR measured range rate error variance corresponding to a percentage error

VAR_{VMIN} onboard constant that represents the minimum RR measured range rate error variance

VAR_{VHF} onboard constant that represents the a priori estimate of the VHF measured range error variance corresponding to a percentage error

VAR_{VHFMIN} onboard constant that represents the minimum VHF measured range error variance

VAR_{β} onboard constant that represents the a priori estimate of the RR measured shaft angle error variance

VAR_{θ} onboard constant that represents the a priori estimate of the RR measured trunnion angle error variance

VHF_{BIAS} bias on VHF range measurement

V_{CON} array that contains conversion factors needed to convert various velocity units to fps

V_E, V_e thrust exhaust velocity

V_G magnitude of velocity-to-be-gained vector

V_{GX}, V_{GY}, V_{GZ} components along the LM X-,Y-, and Z-body axes, respectively, of the velocity to be gained

V_{gx}, V_{gy}, V_{gz} components along the LM X-,Y-, and Z-body axes, respectively, of the total LM velocity to be gained during the burn

V_H a. horizontal velocity component

b. desired horizontal velocity at insertion

$V_{H,I}$ horizontal velocity component of the active vehicle

V_R horizontal component of the present LM velocity

V_{Ra} horizontal component of LM velocity vector parallel to the CSM orbit, at present time in orbit insertion mode, at t_{ig} in CSI or CDH mode

V_{hf} required horizontal component of LM velocity parallel to CSM plane for the current guidance mode

V_{IGV} desired velocity magnitude at the DPS full throttle position

V_{IH} desired horizontal velocity of active vehicle at CDH

V_{IV} desired radial velocity of the active vehicle at CDH

V_{I1} velocity of the active vehicle CSI

V_{po} predicted velocity to be gained at CDH; computed in the CSI mode only in AGS

V_{py} predicted LM out-of-CSM plane velocity at t_{ig} for CSI, CDH, and TPI; current for orbit insertion

V_t total change in velocity required for a direct transfer

V_{TRIM}	magnitude of the trim maneuver
$V_{v,I}$	vertical velocity component of the active vehicle
$V_{V,J}$	vertical velocity component of the passive vehicle
V_{Y}	component of LM velocity in direction perpendicular to the CSM orbit
V_{YO}	same as V_{Y}
V_{1}	ΔV of CSI maneuver
V_{10}	ΔV of previous guess at the ΔV of CSI
V_{2FG}	desired target velocity for the velocity-following mode
$\overline{V}$	general symbol for velocity vector
$\overline{V}_{A}$	actual velocity vector
$\overline{V}_{A,I}$	actual velocity vector of active vehicle
$\overline{V}_{A,J}$	actual velocity vector of passive vehicle
$\overline{V}_{AGS}$	AGS estimated LM velocity vector
$\overline{V}_{AVG}$	velocity input to subroutine AGSAG to be integrated
$\overline{V}_{C}$	CSM velocity vector
$\overline{V}_{CON}$	velocity of the conic vector in ENCKE
$\overline{V}_{CSM}$	CSM velocity vector
$\overline{V}_{DG}$	desired velocity target vector in the descent guidance
$\overline{V}_{E}$	estimated state velocity

$\overline{V}_{E,I}$	estimated velocity vector for the active vehicle
$\overline{V}_{E,J}$	estimated velocity vector for the passive vehicle
$\overline{V}_F$	predicted braking velocity for direct transfer
$\overline{V}_G$	velocity to be gained
$\overline{V}_I$	velocity of the active vehicle at the point of the first maneuver of a Lambert sequence
$\overline{V}'_I$	see $\overline{V}_t$
$\overline{V}_{IG}$	velocity vector at ignition time
$\overline{V}_{I1}$	active vehicle velocity at CSI
$\overline{V}_{I2}$	active vehicle velocity at CDH
$\overline{V}'_{I2}$	desired active vehicle velocity after CDH
$\overline{V}_i$	updated velocity vector in subroutine ELPRE
$\overline{V}_J$	velocity vector of the passive vehicle
$\overline{V}_{JC}$	velocity vector of the passive vehicle at CDH phase match
$\overline{V}_{J1}$	velocity vector of the passive vehicle at CSI
$\overline{V}_{J2}$	velocity vector of the passive vehicle at CDH
$\overline{V}_{J3}$	velocity vector of the passive vehicle at TPI
$\overline{V}_{LC}$	relative velocity vector of LM with respect to the CSM
$\overline{V}_M$	velocity vector of the monitoring computer's state
$\overline{V}_N$	present velocity vector in subroutine AGSAG

$\overline{V}_0$ a. input velocity vector to the routines LVLHDX and PMATRX

b. initial velocity vector in ENCKE

c. velocity vector to be updated by subroutines OPS and ELPRE

$\overline{V}_0^*$ saved value of $\overline{V}_0$ in ENCKE

$\overline{V}_P$ velocity vector in platform coordinator

$\overline{V}_{PGS}$ velocity vector of the PGNCS state vector

$\overline{V}_{PP}$ velocity vector in platform coordinates which has a magnitude equal to the ground speed

$\overline{V}_R$ rotated velocity vector

$\overline{V}_{REL}$ actual velocity vector of CSM relative to the LM

$\overline{V}_{ROT}$ velocity of the spacecraft in selenocentric coordinates caused by lunar rotation

$\overline{V}_T$ desired vector after the first maneuver in a Lambert transfer

$\overline{V}'_T$ same as $\overline{V}_T$

$\overline{V}_{T,JPHASE}$ input velocity target vector for each phase of descent

$\overline{V}_1$ horizontal unit vector in transfer plane

$\overline{V}'_1$ unit vector normal to $\overline{U}$, and parallel to the CSM orbital plane at t_{ig} in CSI, CDH, and TPI; at present in orbit insertion

$\overline{V}_2$ a. unit vector horizontal at rendezvous point and in transfer orbital plane

b. also as above except retarded by J^4 in transfer orbital plane

$\overline{V}_5$ predicted LM velocity vector at TPI

W_{AFTER}	vehicle weight after a maneuver
$W_{A,I}$	actual weight of active vehicle
W_{AT}	table of vehicle weight at each thrust phase
W_{BO}	vehicle weight at main engine burn out
W_E	estimated vehicle weight
W_{LM}	LM weight
W_O	vehicle weight at the beginning of a maneuver
W_T	vehicle weight at the beginning of each thrust phase
$\dot{W}$	weight flow
$\dot{W}_{ENG}$	weight flow used in subroutines SPLIT and START
$[W]$	navigation filter W-matrix
$[W_B]$	partitioned W-matrix used in the propagation routine ENCKE
$[W_{ICOM}]$	W-matrix of the I_{COM} computer
$[W_\ell]$	ℓth 3 by 3 matrix generated by partitioning the 9 by 9 W-matrix and indexing the partition by rows
$[W_{UNL}]$	initial W-matrix for an unknown landmark
W_{CY}	Y-component of $\overline{W}_C$
W_{1Y}	Y-component of $\overline{W}_1$
$\dot{W}$	weight flow table for the thrust profile
$\overline{W}_C$	unit vector normal to CSM orbital plane
$\overline{W}_D$	K_D dimensional vector of the initial diagonal values of the W-matrix

$\bar{W}_\ell$ ℓth three-dimensional column vector of the W-matrix; the column vectors are indexed by rows

$\bar{W}_1$ unit vector perpendicular to transfer orbital plane

X general symbol for a six-dimensional state vector

XREAD1 first record format on data unit 13; consists of flags that define the type of maneuver statistics that are to be produced

XREAD2 second record format on data unit 13; consists of reference states and comments to be printed

XREAD3 third record format on data unit 13; consists of sample event time, sample actual stater, sample estimated stater, targeted $\Delta\bar{V}$, and burn statistics

XREAD4 fourth record format on data unit 13; consists of total burned ΔV, braking $\Delta\bar{V}$, and extra orbital parameters for processing

XTRA an array of ten orbital parameters for which statistical processing is desired

X_A actual state vector

X_{AGS} AGS estimated LM state vector

X_E estimated state vector

$X_{E,I}$ estimated state vector of the active vehicle

X_{ES} estimated state vectors that result from sampling of MSFN and onboard tracking uncertainty matrices

X_{J6} same as J^6

X_{LS} landing site position vector

X_{LSA} actual landing site position vector

X_{LSE} estimated landing site position vector

X_{LSR} reference landing site vector

X_R reference state vector

$\overline{X}$ symbol used in subroutine INPUT to represent the input state vector; it may lie in Cartesian coordinates, classical elements, or spherical elements

$\overline{X}_{DD}$ target LM insertion velocity vector

$\overline{X}_R$ rotated state vector in TRANCO

x a. independent variable in the regula-falsi iteration in REGFAL

b. universal variable in subroutine UNIVAR

x_i a. array of samples of the parameter X for which a cumulative distribution table is desired

b. array of random numbers generated by subroutine RANNO

X_{NOM} nominal value of the variable X

X_{K12} same as $K_1^{\,2}$

X_N the nth value for the quotient of the universal variable X and the position vector magnitude

$\overline{X}_b$ unit vector along LM X-body axis

$\overline{X}_{bD}$ unit vector along desired pointing direction in steering

y a. out-of-plane position component

b. yaw angle in subroutine START

c. component of LM vector in the direction perpendicular to the CSM vector in AGS

Y_{DD} desired out-of-plane position at insertion

Y_{ERR} error in the dependent variable y in subroutine ITRATE

Y_i	array of random numbers uniformly distributed on the interval [0, 1]
Y'	previous value of Y
$\dot{Y}$	out-of-plane velocity component
$\ddot{Y}_d$	desired out-of-plane acceleration
$\dddot{Y}_d$	desired derivative of the out-of-plane acceleration
$\overline{Y}_b$	unit vector along LM Y-body axis
$\overline{Z}_b$	unit vector along LM Z-body axis
$\overline{Z}_{bD}$	unit vector along range vector which commands desired pointing direction in Z-axis steering
$\overline{Z}_{bX}$	actual Z-body direction in inertial reference frame
$\overline{Z}_i$	navigation filter vector used in the intermediate computations
$[Z]$	matrix formed by the three $\overline{Z}_i$ vectors
α	reciprocal of the semimajor axis in subroutine CONIC
$\overline{\alpha}^2$	navigation filter variance
α_E	semimajor axis of CSM orbit
α_i	semimajor axis of LM orbit
α_L	desired semimajor axis of LM orbit after burn
α_N	ratio of magnitude of initial position vector to semimajor axis
α_{TOL}	tolerance on the phase angle convergence in COMDKI
β_A	actual RR shaft angle
β_E	estimated RR shaft angle

β_M measured RR shaft angle

γ
a. flight-path angle in subroutine OUTPUT

b. error in phase angle convergence in CFP

c. intermediate variable in the W-matrix update routine DINCP2

d. inertial flight-path angle as measured from vertical in subroutine UNIVAR

γ_0 value of CFP phase angle convergence error on the previous iteration

ΔE_i difference in the final and initial eccentric anomalies in subroutine ELPRE

ΔH_{CDH} differential height at CDH

ΔH_D desired differential height at the second maneuver of the modified SOR logic

ΔH_{TPI} desired differential height at TPI

Δh differential height

Δh_D desired differential height at CDH

Δh_{MAX} maximum differential height between two orbits

Δh_{MIN} minimum differential height between two orbits

ΔM_i difference in the final and initial mean anomalies

ΔP change in P for next iteration in p-iterator portion of AGS

ΔR_M measured relative range

$\Delta\dot{R}_E$ estimated relative range rate

$\Delta\dot{R}_M$ measured relative range rate

Δr difference in the radial distances of the CSM and LM at phase match at CDH

Δr_1 a. difference between the X-components of measured range and the computed range

b. the difference between the measured range rate and the computed range rate

ΔT a. integration step size

b. guidance cycle length

ΔT_B burn time for a maneuver

ΔT_{BI} half the burn time for a maneuver

ΔT_{BP} table of phase times for the thrust profile

ΔT^*_{BP} stored table of thrust profile phase times

ΔT^*_{BPA} equivalent to ΔT^*_{BP}

ΔT_{BRAKE} first guess at the braking phase duration

ΔT_{BURN} time spent in a thrusting maneuver; includes ullage

$\Delta T'_{BURN}$ ΔT of the current thrust maneuver from ullage ignition to the current guidance cycle

ΔT^*_{BURN} ΔT between ullage ignition and main engine cutoff

ΔT_{CA} time tolerance used to define convergence of the closest approach iteration of TPASS

ΔT_{CL} desired maneuver line time bias from the apsidal crossing

ΔT_{ENG} thrust profile phase timer set in subroutine SPLIT and input to subroutine START

ΔT_{GO} incremental adjustment to T_{GO} in the descent guidance logic

ΔT_{IG} incremental adjustment to the PDI calculation of the preignition phase

ΔT_{INT} integration step size

ΔT_{MARK} time increment between marks

ΔT_{MAX} maximum time step allowed in elevation angle search

ΔT_{P} ΔT of the current thrust phase

ΔT_{PASS} minimum acceptable ΔT of a CSM pass over a landmark

ΔT_{SAVE} table of ΔT's of CSM passes over the landmark to be tracked

$\Delta T_{TAIL-OFF}$ duration of burn tail-off phase

$\Delta T^{*}_{TAIL-OFF}$ stored value for $\Delta T_{TAIL-OFF}$

ΔT_{TEST} minimum acceptable time between the optical tracking of two landmarks

ΔT_{TO} duration of burn tail-off phase

ΔT_{TOL} minimum time between the maneuver line threshold time and the initial maneuver line

ΔV magnitude of velocity vector increment

$\{\Delta V_k\}_{k=1}^{N}$ table of samples of total ΔV required for the mission profile

ΔV_{MAN} total ΔV of a maneuver

ΔV_{MIN} minimum ΔV required during a guidance cycle for the guidance to consider the engine to be burning

ΔV_{REF} reference value for the total ΔV required for the mission profile

ΔV_{TO} tail-off ΔV

ΔV_{TOT} total ΔV of a maneuver

ΔV_{TOTAL}	total ΔV of a maneuver
ΔV_1	ΔV remaining to be burned by the main engine before cutoff
ΔX_A	actual relative state vector
ΔX_E	estimated relative state vector
Δx	a. step size in the computations of the cumulative distribution table of subroutine CDFUNC b. increment in the independent variable x computed in REGFAL c. increment in the universal variable x in subroutine CONIC
ΔZ	increment in the general independent variable in subroutine ITRATE
$\Delta\Delta h$	maximum variation in the differential height between two orbits
$\Delta\alpha$	phase error at TPI in subroutine COMDKI
$\Delta\theta$	phase error at TPI in subroutine PMISS
$\overline{\Delta IG}$	integrated gravity acceleration vector per 2-second computer cycle
$\Delta\overline{Q}_S$	sum of sensed velocity increments $\Delta\overline{q}_p$
$\Delta\overline{q}_p$	sensed velocity increments
$\overline{\Delta q}_s$	identical with $\overline{\Delta Q}_S$
$\Delta\overline{R}$	vector increment to a radius vector
$\Delta\overline{R}_A$	actual relative position vector
$\Delta\overline{R}_E$	estimated relative position vector
$\Delta\overline{V}$	incremental velocity vector

$\Delta\overline{V}_A$	a. actual incremental velocity vector b. actual relative velocity vector
$\Delta\overline{V}_{AGS}$	incremental velocity sensed by the AGS accelerometers
$\Delta\overline{V}_B$	accelerometer ΔV read bias
$\Delta\overline{V}_{BR}$	incremental velocity vector in basic reference coordinates
$\Delta\overline{V}_{CDH}(LV)$	required CDH ΔV in local vertical coordinates
$\Delta V_{CSI}(LV)$	required CSI ΔV in local vertical coordinates
$\Delta\overline{V}_E$	a. estimated incremental velocity vector b. estimated relative velocity vector
$\Delta\overline{V}_{EX}$	incremental velocity for an external ΔV maneuver
$\Delta\overline{V}(LV)$	incremental velocity in local vertical coordinates
$\Delta\overline{V}_{LV}$	equivalent to $\Delta\overline{V}(LV)$
$\Delta\overline{V}_{N1}$	the $\Delta\overline{V}$ of the first maneuver of a DKI sequence
$\Delta\overline{V}_{N2}$	the $\Delta\overline{V}$ of the second maneuver of a DKI sequence
$\Delta\overline{V}_{N3}$	the $\Delta\overline{V}$ of the third maneuver of a DKI sequence
$\Delta\overline{V}_P$	accelerometer sensed $\Delta\overline{V}$ in platform coordinates
$\Delta\overline{V}_{PGS}$	incremental velocity sensed by the PGNCS accelerometers
$\Delta\overline{V}_S$	sensed ΔV during a guidance cycle
$\Delta\overline{V}_{TPF}$	incremental velocity required for velocity match at rendezvous
$\Delta\overline{V}_u$	unit vector directed along $\Delta\overline{V}_{VEC}$

$\Delta\overline{V}_{VEC}$ incremental velocity targeted for a maneuver

$\Delta\overline{V}_1$ ΔV of the first maneuver in a maneuver sequence

$\Delta\overline{V}_2$ ΔV of the second maneuver in a maneuver sequence

$\Delta\overline{X}_3$ correction vector to a landmark position vector

δ minimum central angle between the radius vector of the spacecraft and that of a landmark

δQ discrepancy between the measured value and estimated value of a tracking parameter

δR measurement bias on range

δR_{VHF} VHF measurement bias on range

$\delta\dot{R}$ measurement bias on range rate

$\delta\overline{R}_A$ error in the estimated relative range vector after a tracking mark has been incorporated

$\delta\overline{R}_B$ error in the estimated relative range vector before a tracking mark has been incorporated

$\delta\overline{R}_C$ correction to the estimated relative range vector

δr a. stable orbit displacement distance

b. difference in the CSM and LM radial distances at phase match at t_{ig} in AGS

δT a. time lag between the target vehicle and the desired chaser position at the time of the second maneuver of the stable orbit sequence

b. guidance cycle increment in the burn simulation routine RUNGA

c. general variable for a time step increment

δT_{ABORT} time from powered descent ignition at which an abort is to occur

δT_{INC} integration interval in the burn simulation routine RUNGA

δT_{STAGE} maximum time from powered descent ignition at which staging is to occur

δT_{START} time increment from the last event to the start of onboard tracking

δT_{US} integration step size during ullage

δT_4 time increment remaining until the end of the current guidance cycle

δU tolerance on U used in placing U in the proper quadrant in the AEG

δV_X current iteration on the N_{C1} maneuver in subroutine COMDKI

δV_1 previous iteration on the N_{C1} maneuver in subroutine COMDKI

δV_2 current iteration on the N_{C1} maneuver in subroutine COMDKI

δX state vector dispersion

δX_E dispersion of the estimated state vector from the actual state vector

$\{\delta X_k\}_{k=1}^{N}$ table of state vector dispersion in the subroutine SAMDIF

δX_{LA} dispersions obtained from sampling the lunar ascent insertion matrix

δX_{LSA} dispersion of the actual landing site off the reference landing site

δX_{LSE} dispersion of the estimated landing site off the actual landing site

δX_S sum of the entries in the table $\{\delta X_k\}_{k=1}^{N}$

δ_A maximum acceptable central angle between the radius vector of the spacecraft and that of a landmark which is to be tracked

$\delta_{SXT_x}, \delta_{SXT_y}, \delta_{SXT_z}$ angular bias error for each axis of the unit vector produced by sextant tracking

$\delta\alpha_1$ phase angle error on the previous N_{C1} iteration in subroutine COMDKI

$\delta\alpha_2$ phase angle error on the current N_{C1} iteration in subroutine COMDKI

$\delta\beta$ RR shaft angle measurement bias

$\delta\beta_E$ onboard navigation filter estimate of $\delta\beta$

$\delta\Delta X_E$ difference between ΔX_A in local vertical coordinates and ΔX_E in local vertical coordinates

$\delta\Theta$ RR trunnion angle measurement bias

$\delta\Theta_E$ onboard navigation filter estimate of $\delta\Theta$

$\delta\mu_i$ sample mean of the differences between the reference value of the parameter x and the sample values of x

δ_2 staying flag in AGS

$\overline{\delta}$ position vector deviation from $\overline{R}_{CON}$, which is the conic position vector in subroutine ENCKE

$\overline{\delta}*$ saved value of $\overline{\delta}$

$\overline{\delta}_i$ equivalent to $\overline{\delta}$ indexed by vehicle number

$\overline{\delta}_S$ saved value of $\overline{\delta}$

$\delta\overline{V}_A$ error in the estimated relative velocity vector after a tracking mark has been incorporated

$\delta\overline{V}_B$ error in the estimated relative velocity vector before a tracking mark has been incorporated

$\delta\overline{V}_C$ correction to the estimated relative velocity vector

$\delta\overline{X}_i$ correction to the estimated state vector being updated

$\varepsilon_x, \varepsilon_y, \varepsilon_z$ angular error for each axis of the unit vector produced by sextant tracking

ε_D desired elevation angle at TPI

ε_{ij} element of the ε-matrix which consists of the error angles for the accelerometer mounting

ε_{MIN} minimum acceptable tracking visibility angle from a landmark to an orbiting spacecraft

ε_T tracking attitude deadband

$\overline{\eta}$ random vector of variable dimension; the vector is generated in such a way that each component has a mean of zero and a standard deviation of 1

Θ a. phase angle

b. true anomaly

c. half the orbital arc traveled by the spacecraft while visible to a landmark (subroutine TPASS)

Θ_A actual RR trunnion angle

Θ_{CANT} engine cant angle

Θ_D desired phase angle at TPI

Θ_f central angle between CSM and LM at t_{ig} in CSI mode of AGS; at present in orbit insertion mode

Θ_{INS} desired true anomaly at insertion

Θ_{LOS} line-of-sight angle of CSM relative to the LM

Θ_M measured RR trunnion angle

Θ_T orbital arc traveled by the spacecraft during a burn

θ_{TPI}	phase angle at TPI
$\theta_x,\theta_y,\theta_z$	platform misalinement angles
$\theta_{x,o};\theta_{y,o};\theta_{z,o}$	initial platform misalinement angles
$\dot{\theta}_x,\dot{\theta}_y,\dot{\theta}_z$	platform gyro drift rates
λ	a. longitude of the spacecraft b. longitude of a landmark
λ_K	longitude of the kth landmark
λ_R	longitude of the landmark for which visibility data are to be computed in subroutine TPASS
$\lambda_{R,0}$	input λ_R in selenographic coordinates (subroutine TPASS)
μ	a. gravitational constant of the central body b. mean of a random variable
μ_E	gravitational constant of the earth
μ_i	sample mean of the ith parameter table processed by subroutine SAMSIG
μ_R	mean of the radius vector dispersions
μ_V	mean of the velocity vector dispersions
μ_2	iteration control counter in subroutine ELPRE
μ_3	p-iteration control counter in AGS
μ_8	ullage counter
$\bar{\mu}$	mean of a random vector
$\bar{\nu}$	deviation of the velocity vector from $\bar{\nu}_{CON}$, which is the conic velocity vector of subroutine ENCKE

$\bar{\nu}_i$ vector $\bar{\nu}_i$ indexed by vehicle number

$\bar{\nu}^*$ same value as $\bar{\nu}$

ρ_{ij} correlation coefficient for the ith and jth components of a state vector

Σ covariance matrix

Σ_A covariance matrix of actual state vector dispersions from the reference state vector

$\Sigma_{A,I}$ Σ_A for the ith vehicle

Σ_{LA} covariance matrix that represents the dispersions of the actual and estimated states at insertion caused by powered ascent

Σ_{LSA} covariance matrix that represents the dispersions of the actual landing site from the reference landing site

Σ_{LSE} covariance matrix that represents dispersions of the estimated landing site off the actual landing site

$\Sigma_{r,c}$ the 6 by 6 matrix in the rth row and cth column of a square partitioning of a 12 by 12 covariance matrix

Σ_{REL} covariance matrix of dispersions to the estimated relative state vector

$\Sigma_{R,K6}$ storage array for 6 by 6 tracking covariance matrices

Σ_T a 12 by 12 covariance matrix of MSFN tracking uncertainties for two vehicles

$\Sigma_{T,K12}$ storage array for 12 by 12 tracking covariance matrices

σ standard deviation

$\left.\begin{array}{l}\sigma B_{SXT_x}, \\ \sigma B_{SXT_y}, \\ \sigma B_{SXT_z}\end{array}\right\}$ standard deviation of the angular bias in each axis of the unit vector produced by sextant tracking

σB_{VHF} standard deviation of the VHF range measurement bias

σ_{AAL} standard deviation of the angular accelerometer misalinement for each axis

σ_{ALN} standard deviation of the initial platform misalinement angles for each axis

σ_{DFT} standard deviation of the platform gyro drift rate for each axis

σ_{DVB} standard deviation of the accelerometer read bias for each axis

σ_F standard deviation of the thrust magnitude

σ_I standard deviation of the specific impulse

σ_i standard deviation of the ith parameter of an array

σ_{ij} element of the ith row and jth column of the covariance matrix $\sum$

σ_{N_R} standard deviation of the RR noise on the range measurement

$\sigma_{N_{\dot{R}}}$ standard deviation of the RR noise on the range rate measurement

σ_{N_β} standard deviation of the RR noise on the shaft angle measurement

σ_{N_Θ} standard deviation of the RR noise on the trunnion angle measurement

σ_R standard deviation of the radius vector dispersions

$\left.\begin{matrix}\sigma_{SXT_x}\\ \sigma_{SXT_y}\\ \sigma_{SXT_z}\end{matrix}\right\}$ standard deviation of the angle noise in each axis of the unit vector produced by sextant tracking

σ_V standard deviation of the velocity vector dispersions

σ_{SR} standard deviation of the bias in the RR range measurement

$\sigma_{S\dot{R}}$ standard deviation of the bias in the RR range rate measurement

$\sigma_{\delta\beta}$ standard deviation of the bias in the RR shaft angle measurement

$\sigma_{\delta\Theta}$ standard deviation of the bias in the RR trunnion angle measurement

τ a. time since last rectification in ENCKE

b. ratio of vehicle weight to weight flow in the LM ascent guidance logic

τ_G time increment, in LM descent guidance, during which velocity errors are to be damped out during the velocity-following mode

τ_S time increment to the termination of the current guidance phase in LM descent guidance

τ^* saved value of τ at insertion

ϕ latitude

ϕ_{ASCENT} powered flight arc of ascent

ϕ_{CSM} travel angle of the CSM during powered ascent

ϕ_D desired phase angle at insertion in a powered ascent

ϕ_K latitude of the kth landmark

ϕ_R latitude of the landmark for which visibility data are to be computed in subroutine TPASS

$\phi_{R,0}$	input ϕ_R in selenographic coordinates (subroutine TPASS)
ϕ_{TLO}	desired phase angle at lift-off in a lunar ascent
ψ	heading angle from north
ψ_P	sine of LM pitch attitude
ψ_Y	sine of desired LM out-of-plane attitude
ω	magnitude of the vector $\bar{\omega}_C$
ω_A	mean motion of the passive vehicle
ω_E	rotation rate of the earth about its polar axis
ω_{MOON}	rotation rate of the moon about its polar axis
ω_P	mean motion of the passive vehicle
ωt	target travel angle during a chaser two-impulse sequence
$\dot{\omega}_I$	apsidal rotation rate of the active vehicle
$\bar{\omega}$	thrust turning rate vector
$\bar{\omega}_C$	commanded thrust turning rate vector
$\bar{\omega}_i$	Kalman filter weighting vector used in the onboard tracking routines
$\bar{\omega}_P$	angular rotation rate vector of the moon in spacecraft platform coordinates
$\bar{\omega}_{SC}$	angular rotation rate vector of the moon in selenocentric coordinates
$\bar{\omega}_{SG}$	angular rotation rate vector of the moon is selenographic coordinates

REFERENCES

1. The Monte Carlo Method, The Method of Statistical Trials: Shreider, Yu. A., Ed.; Tee, G. J., Translator. Pergamon Press (Oxford and New York), 1966.

2. MIT Instrumentation Lab: Guidance Systems Operations Plan for Manned LM Earth Orbital and Lunar Missions Using Program Luminary 1A (Luminary Rev. 099). MIT Instrumentation Lab, June 1969.

3. MIT Instrumentation Lab: Guidance Systems Operations Plan for Manned CM Earth Orbital and Lunar Missions Using Program Colossus 2 (Manche 45, Rev. 2), Section 5, Guidance Equations (Rev. 6). MIT Instrumentation Lab, April 1969.

4. TRW Systems Group: LM AGS Programmed Equations Document, Flight Program 6. NAS 9-8166, Document 111 76-6041-TO-00, April 1969.

5. Braley, D. M.: Revision 1 to the User's Manual for the Apollo Guidance Analysis Statistical Trials Program. MSC IN 69-FM-95, April 25, 1969.

6. Nolley, J. W.: Revision 1 to the Error Source Data for Dispersion Analysis. MSC IN 69-FM-160, June 9, 1969.

7. Anderson, T. W.: An Introduction fo Multivariate Statistical Analysis. John Wiley and Sons, Inc., New York, 1958.

8. Kenyon, E. J.: Logic for Earth Orbital AEG in the Apollo Real-Time Rendezvous Support Program. MSC IN 68-FM-119, May 21, 1968.

9. Kenyon E. J.; Sullivan W. A.; and Reini, W. A.: Logic and Equations for the Real-Time Computation of a Satellite Ephermeris About the Moon in Terms of a Specialized Potential. MSC IN 68-FM-12, January 15, 1968.

10. Flanagan, P. F.: RTCC Requirements for Missions E, F, and G: Greenwich Hour Angle Formulation for the Predictor. MSC IN 68-FM-53, February 26, 1968.

11. Simms, R. E.: Logic for the Real-Time Optimization of the Maneuvers for an M = 3 Rendezvous. MSC IN 65-FM-168, December 20, 1965.

12. Alexander, J. D.: Two-Impulse Rendezvous Technique Incorporating Earth Perturbations and Drag. MSC IN 65-FM-27, July 27, 1965.

13. McDonough, R. K.; and Sullivan, W. A.: Logic and Equations for the Real-Time Computation of AS-207/208 Insertion Elements and Specialized Orbital Maneuvers. MSC IN 66-FM-116, October 9, 1966.

14. Regelbrugge, R. R.: Logic for the Real-Time Computation of the Docking Initiation Table. MSC IN 64-FM-59, November 25, 1964.

15. Sullivan, W. A.: Logic and Equations for the Real-Time Computation of the Lunar Module Descent Planning Table. MSC IN 67-FM-68, May 9, 1976.

16. Simms, R. E.: Optimization of the Maneuvers for an M = 3 Rendezvous. MSC memo 66-FM6-16, February 15, 1966.

17. Laycock, G. B.: Analytical Orbit Prediction Program. TRW 66-FMT-481, January 15, 1967.

18. MIT Instrumentation Lab: CMC Guidance and Navigation Equations, Lunar Landing Mission GSOP Preliminary, Section 5. MIT Instrumentation Lab, June 1967.

19. Kenyon, E. J.; Sullivan W. A.; and Reini, W. A.: Logic and Equations for the Real-Time Computation of a Satellite Ephemeris Abort the Moon in Terms of a Specialized Potential. MSC IN, Change 1, 68-FM-12, May 12, 1969.

20. Roth, H. L.; and Escobal, P. R.: Transformations Involving the Selenographic Coordinate System. TRW 3400-6045 RU 000, June 14, 1965.

21. TRW Systems Group: Apollo Reference Mission Program. TRW no. 111 76-8146-RO-00, NAS 9-8166, February 24, 1969.

22. UNIVAC 1108 MATH-PACK, Programmer's Reference. Sperry Rand Corporation, 1967.

23. Abramowitz, M; and Stegun, I.: Handbook of Mathematical Functions. U. S. Department of Commerce, Nat. Bur. Standards, Applied Math Series, no. 55, 1965.

24. Ratcliff, W. D.: A Non-Iterative Method for the Determination of Actual Apogee and Perigee Radii. MSC IN 67-FM-202, December 27, 1967.

25. McHenry, E. N.: Logic for Real-Time Computation of Lunar Concentric Rendezvous Maneuvers. MSC IN 68-FM-35, February 9, 1968.

26. Rothrock, J. E.: An Analytical Method of Simulating the Sensed Velocity Increments Measured by the Inertial Measurement Unit of a Guided Spacecraft. MSC IN 67-FM-139, September 28, 1967.

27. Moore, A. J.; and Reini, W. A.: Determination of Orbital Arrival Times and the Relative Conditions Between the Agena and Gemini Spacecraft Vehicles at an Orbital Comparison Point. MSC IN 64-FM-55, November 23, 1964.

28. Sullivan W. A.: Logic and Equations for the Real-Time Computation of the Lunar Module Launch Window and Recommended Lift-Off Time. MSC IN 67-FM-1, January 6, 1967.

29. McHenry, E. N.: A Scheme for Rapid Evaluation of the Gemini Network. MSC memo, December 31, 1963.